JobSeeker: Easy Job Hunting

Amalesh Arivanan, Emory Lu, Mahirah Samah
Texas A&M University, College Station, Texas
{amalesh.arivanan-22, qianlu, mahirah.samah}@tamu.edu

ABSTRACT

Many existing job-hunting tools lack personalization, overwhelm users with excessive information, and are not intuitive for those new to the job market. To address these challenges, we present *JobSeeker*, an intelligent job recommendation system specifically designed for users with computer science, engineering, and related technical backgrounds. *JobSeeker* provides personalized and relevant job suggestions while prioritizing user privacy and avoiding information overload. Our approach involves several key components: maintaining an up-to-date database of job postings, developing a conversational AI chatbot to interact with users and refine job recommendations through continuous feedback, designing intuitive user interfaces, and organizing an efficient workflow. We also outline an evaluation plan for assessing the system's effectiveness and discuss potential directions for future work.

Keywords

Intelligent User Interface; Interactive Dashboard; Smart System; Job Recommendation System; Chatbots; Conversational AI.

1. INTRODUCTION

Modern job recommendation systems, while helpful, often lack the personalization that many job seekers need. Platforms like LinkedIn and Indeed host hundreds of thousands of jobs, allowing users to filter results by generic criteria such as job type, location, salary, etc. However, these generic filters fail to address more personalized preferences, such as company policies, benefits, work culture, and more advanced criteria are almost always difficult to set in popular job websites.

To address these gaps, we introduce *JobSeeker*, a personalized job recommendation system designed specifically for users with computer science, engineering, and related technical backgrounds. Our system aims to make job searching more personalized and more time-saving. Using artificial intelligence, *JobSeeker* communicates with users to remember their preferences over time and display jobs that the user feels are more suited to their needs.

JobSeeker's tasks are difficult to achieve due to the usage of Chatbots and the complexities of job searching. Since job searching has a lot of factors that a user might have preferences for, an artificial intelligence algorithm to streamline this process may require many operations, which results in a considerable amount of effort and large overhead. JobSeeker is aiming to use Natural Language Processing (NLP) to implement a Chatbot that takes note of user preferences and applies it to filter the jobs that are displayed on the application.

2. DISCUSSION OF ISSUES

Our intelligent job recommendation system faces several challenges across cognitive, social and technology/design aspects. From a **cognitive perspective**, information overload can overwhelm users, so we need to refine the recommendation process to avoid flooding them with irrelevant or excessive job suggestions. A key issue, seen on platforms like LinkedIn, is ensuring that job recommendations are relevant, and avoiding completely irrelevant jobs being suggested. On the **social side**, we must prioritize protecting users' privacy, especially as we implement the interactive chatbot feature. Users will likely share personal information during conversations, so special attention must be paid to safeguarding their data. Additionally, many people find automated job recommendations impersonal, so our system should incorporate a "human touch", even when interacting with AI. This requires implementing an AI model that can process natural language and make interactions feel more personalized. On the **technology and design side**, keeping job recommendations up-to-date can be challenging, given the rapid changes in the job market. Our system must stay aligned and continue delivering timely and relevant recommendations. Finally, balancing a user-friendly interface with clarity and detailed information is essential to providing a positive user experience. By addressing these interconnected issues, we can create an effective and user-focused job recommendation platform.

3. APPROACH

To help users find valuable and legitimate job recommendations, we are planning to divide our work into different parts.

○ 3.1 Database

Our first step is to design and manage a database that organizes job postings efficiently. Considering the complexity of maintaining a large database, we focus specifically on job postings for users with computer science, engineering, and other related technical backgrounds. This reduces redundancy and ensures that the database aligns with the needs of our user base.

In addition, we carefully consider the key data points to include in the database. These include job roles, required skills, preferred locations, and other relevant factors, all chosen with a user-centric perspective. By prioritizing these elements, we aim to ensure that our database addresses the specific concerns and preferences of our users, creating a foundation for accurate and relevant job recommendations.

○ 3.2 Conversational AI Design

To provide personalized job recommendations, we employ a conversational AI chatbot. This chatbot interacts with users to capture input on their job preferences, including desired roles, locations, work hours, and other important information and criteria. Then the chatbot searches the database and presents a list of personalized job recommendations.

A feedback loop is implemented. As users engage with the chatbot and provide additional feedback, the AI dynamically refines its recommendations to align better with users' preferences. This iterative process ensures that job suggestions become increasingly relevant and personalized over time.

○ 3.3 User-Friendly Workflow

Creating a user-friendly system requires careful consideration of the user's workload and experience. The overall workflow must be designed to streamline tasks, ensuring that users are only presented with essential actions to avoid unnecessary complexity.

The number of user interfaces should be minimized, and each interface should include only the necessary elements to prevent confusion and cognitive overload. Interfaces should be intuitive and straightforward, with clear navigation and functionality. When needed, tips or hints should be provided to guide users effectively and enhance their understanding of the system. This approach ensures an efficient user experience.

○ 3.4 Testing and Iteration

To ensure the effectiveness and usability of the AI chatbot and matching algorithm, they will be continuously tested with a sample user base. This iterative approach allows us to identify areas of improvement and refine the system to better meet user needs.

Testing will involve multiple stages, starting with internal evaluations where the system is assessed for functionality, accuracy, and responsiveness. In this phase, we will focus on technical performance metrics such as the accuracy of job matches, the efficiency of the chatbot's natural language processing, and the system's ability to handle user interactions seamlessly. Any identified issues, such as incorrect job recommendations or delays in chatbot responses, will be addressed before involving external users.

Once the system meets our expectations, we will proceed with user testing involving a diverse group of participants who represent the target audience, including users from computer science, engineering, and related technical fields. During these sessions, users will interact with the chatbot, rate job recommendations, and navigate the platform. Feedback will be collected through surveys, interviews, and usage analytics to evaluate key aspects such as user satisfaction, ease of use, and the relevance of job matches.

Based on user feedback, we will implement improvements over the following aspects:

- ***User Interface:*** Adjustments will be made to ensure the platform remains intuitive and visually appealing. Feedback on navigation, design clarity, and feature accessibility will guide changes to enhance the user experience.
- ***Job Recommendations:*** The matching algorithm will be fine-tuned to improve the relevance of job suggestions. Insights from user ratings and interactions will help refine the algorithm to better align with user preferences and needs.
- ***Chatbot Performance:*** Natural language processing capabilities will be improved to make interactions with the chatbot more natural and engaging. Misunderstandings or limitations in the chatbot's responses will be addressed to build trust and usability.
- ***System Performance:*** Backend optimizations will ensure that the platform remains responsive and capable of handling large volumes of data and interactions without performance degradation.

○ 3.5 Future Enhancements

Future developments will include:

- ***Interactive Dashboards:*** To track job applications, interview schedules, and progress.
- ***Autofill Tool:*** To streamline the application process by auto-filling repeated information.
- ***Job Boards Integration:*** Connecting with major job boards to keep recommendations current.
- ***AI-Powered Resume Reviews:*** An AI tool to review resumes and ensure they align with job descriptions for better matching.

4. PRIOR AND RELATED WORK

Several studies have explored AI-driven job recommendation systems that leverage user data to provide personalized suggestions. Yang and D. [10] propose an intelligent recommendation algorithm for college students, utilizing students' skills, qualifications, and interests to match them to job opportunities based on high-demand knowledge areas and changing industry sectors (Figure 1). Similarly, Schlippe and B. [6] present *Skill Scanner*, an NLP-based AI model that extracts and clusters job seekers' skills, aligning them with job postings, emphasizing the importance of skill matching.

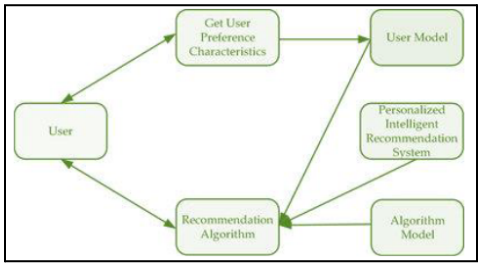

Figure 1: Process of Intelligent Recommender System for the Job Market [10].

Conversational agents have also been explored for job recommendations. Bellini et al. [1] designed *Guapp*, a chatbot that progressively refines job preferences through natural language interaction, offering tailored recommendations based on user input. GUapp utilizes a content-based recommender system driven by Latent Dirichlet Allocation (LDA) to match users' profiles with suitable job opportunities, as illustrated in Figure 2. The system incrementally builds and refines user profiles based on natural language inputs, LinkedIn integration, and user interactions. The platform also features a conversational agent that facilitates dynamic, natural language-based interactions, enabling users to progressively tailor job recommendations.

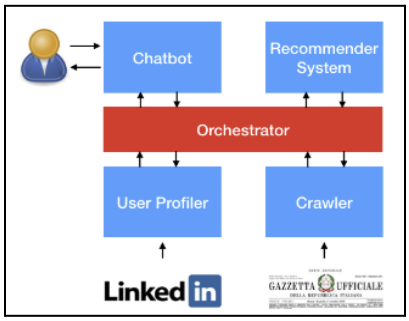

Figure 2: GUapp's Architecture [1].

Suresh et al. [7] and Nguyen et al. [5] both developed chatbots that provide personalized career advice, leveraging NLP to understand user inquiries and suggest relevant career paths,

resources, and job listings (Figure 3). Several existing open-source chatbot frameworks such as ChatterBot [3], Tensorflow-Chatbot [2], and chatbot-ui [9] provide foundational tools for developing conversational agents.

User experience in chatbot interactions is another key area of research. Islind et al. [4] focus on improving the design of conversational agents by enhancing their human-like qualities, exploring user perceptions of chatbot personality and comprehension. Practical interface design techniques, as discussed by Tymofiyeva [8], offer further guidance on creating intuitive and user-friendly chatbot interactions.

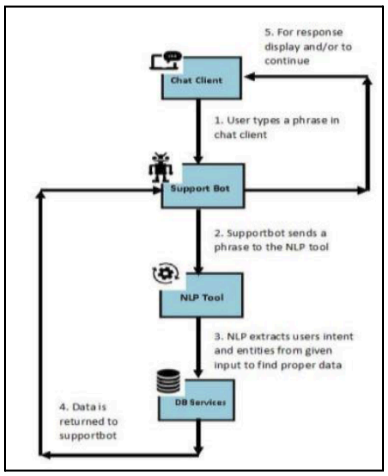

Figure 3: The Working flow of SupportBot [7].

5. DESIGN

○ 5.1 Chatbot - Personalized Interaction

The chatbot interface is designed with a familiar text messaging format (Figure 4), offering simplicity and ease of use. This approach leverages users' familiarity with common messaging platforms, including text boxes, send buttons, and scrolling features, etc. This will help lower the learning curve the users may experience with our platform. By following a familiar interface pattern, users can intuitively understand how to respond and engage with the chatbot, making the interaction feel more approachable and less intimidating. The conversational interface will allow the user to feel more natural in terms of communication with the chatbot and also may help harbor trust in the chatbot. This is important because the user needs to trust that the chatbot will present job postings that are similar to the user's requirements and that it will present genuine job postings.

○ 5.2 Job Recommendation Board

Our old design of the job recommendation board is a page where job postings are shown sequentially, and the user scrolls down to view more job recommendations (Figure 5). This design is intuitive and familiar to the user as many web pages use a similar layout. Many web pages also use lists to display information, which helps this application be more attuned to the user's experiences. Moreover, this design allows the user to focus on individual job postings one at a time. The simplicity of the layout ensures that users can easily view details and decide whether to apply for the position.

A disadvantage of this design is that since it has only the job posting information and the link to apply, it might be difficult for users to keep track of their applications. There are no features that automatically mark which jobs the user has applied to. It can also be difficult when there is a large number of job postings, for which the user might find themselves scrolling back and forth. This can lead to confusion or even missed opportunities.

Figure 4: An example of the Chatbot interface

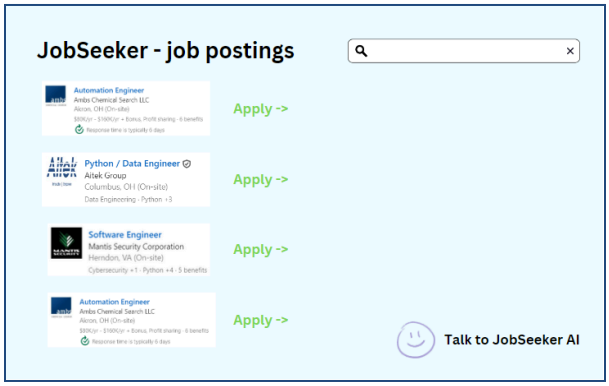

Figure 5: An example of the job application dashboard.

To enhance the job recommendation process, our new design of chatbot collects user feedback on job recommendations through a simple thumbs-up and thumbs-down rating system. Users can rate each job recommendation directly within the chatbot interface, providing valuable input for refining future suggestions. Recommendations that receive positive feedback are saved to the user's profile for easy access and can be revisited later. Additionally, the chatbot utilizes these ratings to continuously improve its recommendation algorithm, ensuring more personalized and relevant job matches over time.

The design for the job board is illustrated in Figure 6. Job postings are displayed sequentially within the chatbot window, eliminating the need for users to navigate to a separate job posting page. To better identify features of jobs and categorize them, we use tags to represent different features and assign them to different jobs. To prevent users from feeling overwhelmed, the chatbot initially shows only a limited number of recommendations that best match users' preferences, with the option to view more if desired. This design is intuitive and familiar, as many web applications use

similar layouts and list-based designs for presenting information. By focusing on one job posting at a time, users can easily review the details and decide whether to apply. The simplicity of the layout ensures that the interface remains user-friendly and accessible, enhancing the overall job search experience.

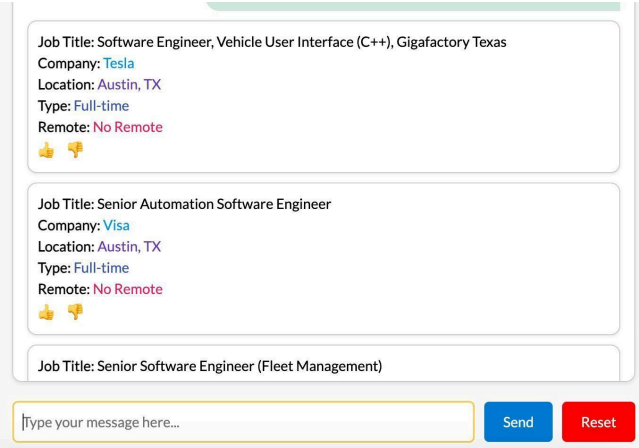

Figure 6: JobSeeker job application dashboard.

○ 5.3 User-Friendly Workflow

The overall design focuses on minimizing the number of UI interfaces to streamline the user experience and reduce cognitive and physical workload (Figure 6). By carefully analyzing the typical user journey, we have optimized the navigation to ensure seamless transitions between interfaces while maintaining clarity and simplicity. As a result, we have reduced the total number of interfaces to five, each serving a specific purpose to address user needs effectively.

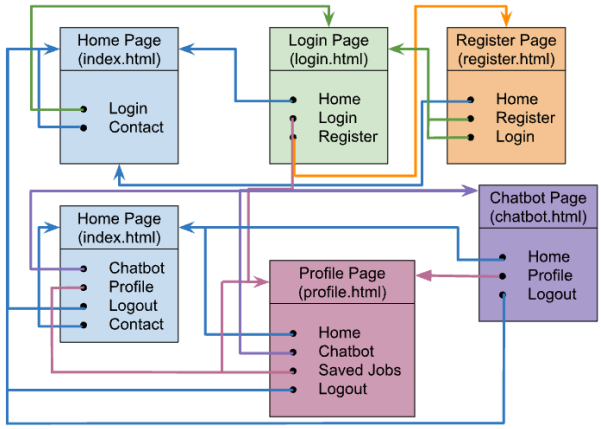

Figure 6. Design of the overall navigation.

Each interface is designed with a minimalistic approach, incorporating only essential UI elements to avoid confusion or cognitive overload. For example, the home interface provides quick access to key features such as contacting and chatbot interactions, while the profile interface allows users to manage their saved jobs and preferences effortlessly. Similarly, the job details interface focuses on presenting clear and concise information about each job posting, with intuitive buttons for applying, saving, or providing feedback.

This design not only aligns with user-centered principles but also enhances accessibility for users who may be less familiar with complex systems. By reducing unnecessary elements and maintaining consistency across interfaces, we ensure that users can navigate the platform intuitively, fostering a positive and efficient job-hunting experience. Furthermore, this approach makes the platform adaptable to future expansions, as new features can be integrated without disrupting the streamlined workflow.

○ 5.4 Color Feature Tags

To better identify job features and organize them effectively, we use **tags** to represent various attributes and assign them to different job postings. Tags help categorize jobs based on specific criteria, such as required skills, benefits, company culture, or work environment, making it easier for users to find jobs aligned with their preferences. As illustrated in Figure 7, these feature tags are grouped into distinct categories, each marked with a unique color for easy visual identification. Users can utilize the filter function to select the categories or specific features they are most interested in, and the corresponding tags will be displayed alongside each job posting with detailed information.

From a computational standpoint, this tagging system is relatively straightforward but costly to implement, as it relies on predefined categories and attributes associated with job data. This simplicity ensures that the system is efficient and responsive, providing users with relevant job insights without significant processing delays. Moreover, by surfacing key job details through tags, users can quickly assess whether a job matches their requirements without delving into lengthy descriptions.

Figure 7. An example of colored feature tags and the filter.

However, we recognize several potential challenges with this approach. It's difficult to manage a database to host predefined categories and attributes, and it's difficult to finalize the final set of tags. Job seekers have diverse and evolving preferences, and no fixed set of tags can fully capture the complexities of every job. If the tags are too little, users can't have enough information; on the other hand, if there are too many tags, it's hard to implement, and users will feel overwhelmed. To address these concerns, we must have a balance by prioritizing the most relevant features, while allowing flexibility to update or expand the tag system based on user feedback.

○ 5.5 Saved Job Lists

We also design to include a feature that allows users to manage a saved list of jobs, providing a personalized space where they can rate, save, or apply for jobs selectively. Users can choose the number of recommendations to be shown, and a brief explanation should be provided to help users understand why the list of recommendations are given.

Figure 8 shows an example of such a design. In the middle section of the interface, recommended jobs are displayed along with checkboxes, enabling users to interact with the recommendations easily. Users can select specific jobs to save for later review, apply directly, or take further action. Beneath each recommendation, buttons provide additional functionality, such as saving selected jobs, applying immediately, or downloading a detailed report file summarizing the chosen recommendations. The ability to download reports is particularly helpful for users who wish to review or share their options offline.

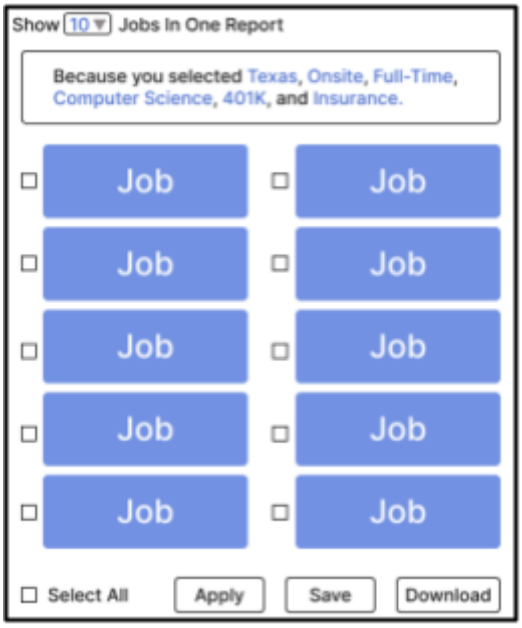

Figure 8. The list of generated job recommendations.

This design has flexibility and simplicity, ensuring that users can easily go through their recommendations while having all essential actions accessible. Additionally, the inclusion of explanatory elements and intuitive controls enhances the overall user experience, making the system both approachable and effective. To ensure continued usability, we plan to gather user feedback on this feature and refine the interface to address any challenges users may encounter. By doing so, we aim to provide a dynamic and user-centered job management tool.

6. IMPLEMENTATION

○ 6.1 Architecture/Data Flow

The dataset utilized in this project consists of job listings primarily focused on software engineering roles within the United States, obtained from publicly available sources. Key attributes include job title, company, location, salary range, remote work options, visa sponsorship availability, and job posting dates (Figure 9). To ensure the integrity and reliability of the job recommendations, extensive data preprocessing was conducted, including data cleaning, normalization, and feature extraction. This process standardized the dataset, enabling efficient filtering and matching algorithms within the chatbot system.

Title	Company	Rating	Work Type	Location	Posting Time	Remote Work	Link	Visa Sponsorship	Job ID

Figure 9. The datasets and the attributes used.

The system follows a structured architecture designed to efficiently process user inputs and provide relevant job recommendations through a seamless interaction between the chatbot interface and the backend. User inputs, such as job title, location, contract type, remote work preferences, and visa sponsorship requirements, are collected through a conversational interface. These inputs are then standardized through a normalization process to ensure consistency, allowing for more accurate data matching (Figure 10). Once normalized, the inputs are passed to the backend, where they are used to filter a preloaded job dataset based on the user's preferences. The filtering logic ensures that jobs matching the user's criteria—such as location, job type, and visa sponsorship—are selected. The filtered job recommendations are then relayed back to the chatbot interface, where they are displayed in a conversational format for easy review. Additionally, the system incorporates a feedback loop, allowing users to provide input on the recommendations and refine their search if necessary. This architecture enables continuous interaction between users and the system, ensuring a smooth flow of data processing and job recommendation delivery. It minimizes user workload through streamlined navigation and interface design, ensuring simplicity and accessibility. Features like saving, liking, and revisiting job recommendations enhance decision-making, while database management prioritizes privacy by collecting only essential data. The design ensures efficient interaction, refined recommendations through feedback loops, and seamless integration of functionality to address user pain points effectively.

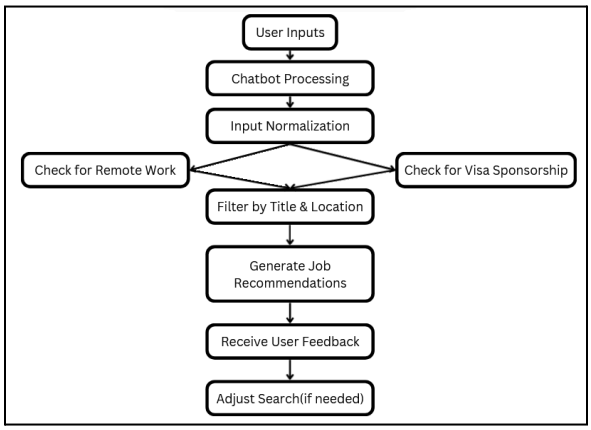

Figure 10. Flowchart of the proposed model.

○ 6.2 Interface

We use the Flask framework in Python 3 and HTML to develop our user interface. We will use AJAX and JavaScript to connect the backend code to the frontend, and PostgreSQL for data storage. The goal is to avoid unnecessary operations and make all pages simple and informative. Through careful designs, we minimized the overall design into four interfaces: a landing page, a job search page, a chatbot interface, and a login page. Specifically, the job search page and the chatbot interface together are the essential components of our interface designs.

6.2.1. Landing Page. As shown in Figure 11, our landing page only contains four interactions, where users can not do anything unless they log in their accounts first. Before logging in, the only thing users can do is to contact the development team. Upon login in, users can additionally (1) navigate back to the home page, (2) talk to the chatbot and start job searching, and (3) manage their profiles pages to check saved jobs.

6.2.2. Chatbot Interface. The chatbot interface (Figure 12) is

designed to look and feel like a real text-messaging app, giving users a familiar and approachable way to interact with the bot. We've set this up using HTML and CSS for the core layout and design, while Python Flask powers the web page itself. To make sure conversations flow smoothly, we're using AJAX and JavaScript to handle the back-and-forth messaging. This lets messages load instantly as the user and bot respond, avoiding any page refreshes and keeping the experience seamless.

Figure 11. JobSeeker landing page.

The interface itself is straightforward. Users have a text box at the bottom where they can type messages, along with a "send" button to submit them. As messages are exchanged, responses from the chatbot appear on the left side of the screen, with the user's messages aligned to the right, much like a text conversation. Users can scroll back to review earlier messages, making it easy to keep track of the conversation.

We also relocated the input boxes and "Send" buttons right below the chat display for better accessibility. For questions that would expect simple inputs, for instance, "yes" or "no", we placed several buttons representing each option, so users can skip the typing. Our goal is to make the interface feel as close as possible to a real messaging app, creating a familiar, user-friendly setup that encourages natural, ongoing interactions with the bot.

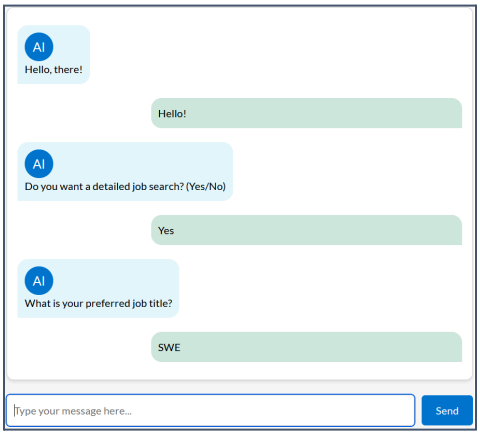

Figure 12: JobSeeker Chatbot interface.

Ultimately, these changes are about making the chatbot easy and enjoyable to use, so users can jump into conversations without a second thought. By bringing in these small design elements, we hope to create an experience that feels fluid, inviting, and true to the look and feel of text messaging.

6.2.3. Login Page. Our login page features a straightforward and minimalist design, similar to the conventional login pages found on many popular websites (Figure 13). This simplicity ensures that users can navigate the page without difficulty or unnecessary distractions. By avoiding unnecessary features, the design prioritizes functionality and ease of use, making the process quick and easy for all users.

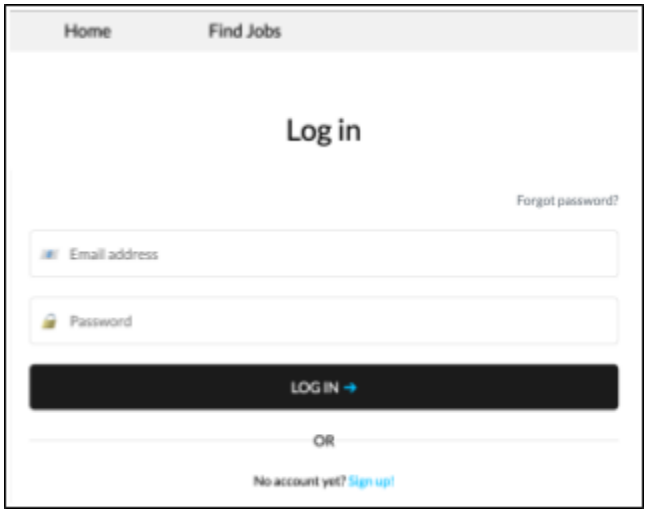

Figure 13. JobSeeker login page.

6.2.5. Profile Page. On this page, users can view their profile details, such as their name, email address, and preferences, ensuring they can easily access and verify their personal information. Additionally, the page provides a section for job postings that users have saved for later review, allowing them to revisit these opportunities at any time without searching again (Figure 14).

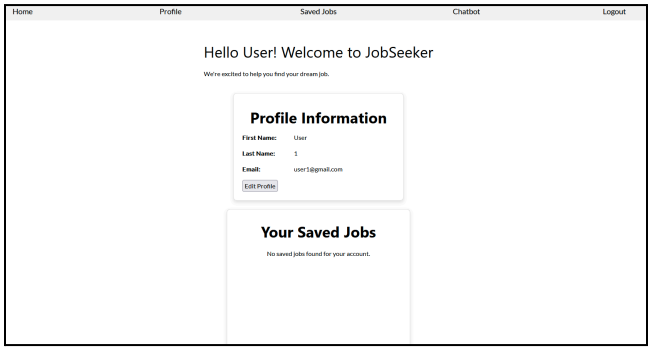

Figure 14. JobSeeker register-user profile page.

6.2.4. Register Page. Our register page, similar to our login page, has a straight-forward design as most conventional sign-up pages from a variety of websites (Figure 15).

Upon user sign up, an entry will be made in the user table in our PostgreSQL database. A unique user ID will be generated for each unique user, and their password will be encrypted when stored into the database. The user table is implemented as shown in Figure 16.

Additionally, when the user is given their job recommendations from the Chatbot interface , the recommendations will be stored in the database in the jobs table. The jobs table consists of the user ID and the corresponding job ID that the Chatbot has chosen for them. Figure 17 shows the jobs table.

Figure 15. JobSeeker register page.

Figure 16: User table in PostgreSQL

The saving & reviewing jobs feature (Figure 6) allows users to save and revisit job recommendations, providing a streamlined and organized way to manage their job search. This functionality enhances the decision-making process by enabling users to like or revisit previously viewed jobs, helping them compare options and make informed choices. By offering a personalized experience, this feature ensures users can efficiently track and review job opportunities that align with their preferences and goals. These data can provide insights into the users' preferences and can be worked on in the future.

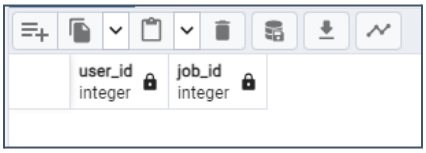

Figure 17: Jobs table in PostgreSQL

Users can view their profile information and access a list of job postings they have saved for later review (Figure 18).

7. EVALUATION

○ 7.1 Outline

The evaluation focuses on measuring the effectiveness of the JobSeeker application in meeting user needs through user satisfaction, usability, and relevance of job recommendations. The primary goal is to ensure that the application delivers accurate and helpful job matches tailored to user preferences while providing a seamless and intuitive interface. Special emphasis will be placed on assessing how well the interface design supports easy navigation and interaction. The insights gained from this evaluation will guide further improvements to enhance the application's usability and user experience, ensuring that the application evolves to meet user demands more effectively.

7.2 Evaluation Type

The evaluation will be conducted as a field study, allowing participants to freely interact with the JobSeeker application in a real-world setting. This approach ensures that user feedback is gathered in a practical and natural context, enabling the team to observe how users navigate the app, use its features, and interact with the intelligent chatbot. Key measures such as task success, usability, and engagement will be analyzed to determine the application's effectiveness in addressing the needs of job seekers.

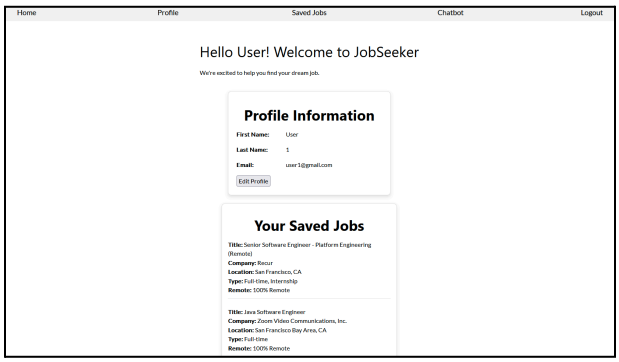

Figure 18: Updated Profile Page

7.3 Participant Selection

Participants for this evaluation will primarily include individuals with a background in computer science/engineering or related technical fields who are actively seeking jobs. To ensure meaningful feedback, participants must possess basic computer literacy and fluency in English, enabling them to engage effectively with the system and provide thoughtful insights into its functionality and usability.

Their familiarity with existing job-seeking platforms like LinkedIn or Indeed will allow them to offer comparative feedback, helping us identify strengths and areas for improvement. By focusing on users with relevant backgrounds and needs, we aim to gather practical and actionable feedback on how well JobSeeker supports their job search process.

To ensure diverse perspectives, we will select participants with varying levels of job-seeking experience, such as recent graduates and mid-career professionals. This diversity will help us evaluate how effectively the system meets the needs of different user groups.

7.4 Activity for Participants

The evaluation process will consist of three key stages to ensure a thorough assessment of user experiences.

Before the interaction, the developer team will introduce participants to the study, providing detailed instructions on the application's features and usage. This stage ensures that participants are familiar with the system and understand the purpose of the evaluation. Any questions about the study or the app's functionality will be addressed to create a comfortable environment for participants.

During the interaction, participants will be observed as they use our website app, with a focus on their ease of use, engagement, and overall preferences. Observations will include tracking how users interact with features like the chatbot, filters, and

recommendation lists, as well as noting any moments of confusion or hesitation.

After completing the interaction, detailed feedback will be collected through discussions and structured surveys. Participants will be asked to reflect on any difficulties they encountered, their overall satisfaction with the app, and how intuitive they found the interface. Specific questions will address whether the app met their expectations, how it compared to other job searching tools they have used, and suggestions for improvement.

7.5 Data Collection & Metrics

Data collection will include both quantitative and qualitative metrics to provide a holistic view of the app's performance. Quantitative data will measure the frequency of clarifications sought by users, the average time spent on the app, and the number of job recommendations users found relevant and selected. Qualitative feedback will capture users' ease of use compared to other job-hunting platforms, suggested improvements, desired additional features, and the key aspects of job searching they value most. Together, these metrics will inform the team about the app's strengths and areas for refinement.

8. CONCLUSIONS & FUTURE WORK

The JobSeeker application presents a promising solution for addressing the complications involved in modern job searching through personalized recommendations and an intuitive user interface. By leveraging conversational AI, adaptive frameworks, and interactive dashboards, the platform aims to create an efficient and user-friendly experience tailored to individual preferences. The JobSeekers application has the potential to extend and explore numerous advancements. To streamline interactions, the chatbot can limit unnecessary exchanges, making personalization quicker and more efficient. Integration with major job platforms like LinkedIn and Glassdoor will expand the database, while an interactive dashboard will provide a centralized view of saved jobs, applications, and progress. An interactive dashboard can be introduced to provide users with a centralized view of their job search activities, including saved jobs and applications, for better organization and decision-making. To support career development, skill assessment tools and personalized training recommendations will help users identify and close skill gaps. A recruiter dashboard will be added to facilitate job postings, application management, and candidate profiling, catering to both job seekers and employers. These improvements aim to create a comprehensive, efficient, and user-friendly job-seeking platform.

Due to the small time frame and the limited number of team members, there are several designs that were not able to be implemented. For example, the colored feature tagging feature, which involves complicated algorithms and managing a fairly large database to sort out all the features then assign colors. Another design is the applying job feature, where users can select multiple jobs and apply at once, and the applied jobs can be seen and managed in the user's profile page. Additionally, the user's profile and the chatbot are overly simplified, there are several features that our team planned to develop, but end up not having enough of the time. For future works, one potential direction is to have the AI resume reading and revising feature, where users can simply upload the resume, the chatbot will read the resume, automatically give job recommendations, meanwhile, the chatbot will revise the resume, so users can use the improved resume to apply for jobs. Another direction is to expand the target users to a larger range, since so far we only manage the database for jobs related to people with computer science/engineering background.

9. WORK PLAN

Tasks are distributed evenly between the three team members. Team members Amalesh and Mahirah collaborate on developing the Conversational AI Chatbot. Specifically, Amalesh focuses on the chatbot design and refining chatbot interaction flow, and Mahirah focuses on the computing of chatbot, as well as database maintenance. They work on designing the chatbot interface while ensuring it effectively captures users' input, like job preferences, experience, and salary expectations, while refining job recommendations based on user feedback. Meanwhile, team member Emory handles both the frontend and backend development, creating a user-friendly interface and integrating the backend infrastructure to support the chatbot. Together, the team will refine the overall workflow, conduct testing, iterate on improvements, and prepare the system for final deployment.

10. REFERENCES

[1] Bellini, V., Biancofiore, G. M., Noia, T. D., Sciascio, E. D., Narducci, F., and Pomo, C. 2020. Guapp: A conversational agent for job recommendation for the italian public administration. *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems* (EAIS) (pp. 1-7). IEEE. DOI=https://ieeexplore.ieee.org/document/9122756.

[2] Buddhika, N. 2017. Tensorflow-Chatbot. *GitHub.* https://github.com/nimeshabuddhika/Tensorflow-Chatbot. Accessed Sep. 18. 2024.

[3] Cox, G. 2020. ChatterBot. *GitHub*. https://github.com/gunthercox/ChatterBot. Accessed Sep. 18. 2024.

[4] Islind, A. S., Óskarsdóttir, M., Smith, S. Á., and Arnardóttir, E. S. 2023. The Friendly Chatbot: Revealing Why People Use Chatbots Through a Study of User Experience of Conversational Agents. *Scandinavian Conference on Information Systems.* DOI=https://www.researchgate.net/publication/372400527.

[5] Nguyen D. , Dinh N., Pham-Nguyen C., Dinh T., and Nam L. 2022. ITCareerBot: A Personalized Career Counseling Chatbot. *Intelligent Information and Database Systems. ACIIDS.Communications in Computer and Information Science*, vol 1716. Springer, Singapore. DOI=https://doi.org/10.1007/978-981-19-8234-7_33.

[6] Schlippe, T. and Bothmer, K. 2023. Skill Scanner. *International Journal of Advanced Corporate Learning* (iJAC), 16(1), 55–64. DOI=https://doi.org/10.3991/ijac.v16i1.34779.

[7] Suresh, N., Mukabe, K., Hashiyana, V., Limbo, A., and Hauwanga, A. 2022. Career Counseling Chatbot on Facebook Messenger using AI. *In Proceedings of the International Conference on Data Science, Machine Learning and Artificial Intelligence* (DSMLAI '21'). *Association for Computing Machinery*, New York, NY, USA, 65–73. DOI=https://doi.org/10.1145/3484824.3484875.

[8] Tymoshchuk, O. 2021. 16 Best Techniques for Creating a User-Friendly Interface. *Geniusee.* https://geniusee.com/single-blog/16-techniques-for-creating-a-user-friendly-interface. Accessed Sep. 18. 2024.

[9] Wrigley, M. 2024. chatbot-ui. *GitHub*. https://github.com/mckaywrigley/chatbot-ui. Accessed Sep. 18. 2024.

[10] Yang, C. and Dong, L. 2024. Intelligent talent recommendation algorithm for college students for the future job market. *Journal of Electrical Systems*, 20(3s), 1822–1832. DOI=https://doi.org/10.52783/jes.17