SKILL/ JOB RECOMMENDER APPLICATION

A PROJECT REPORT

Submitted By

KANIMOZHI A	**511319104032(Team Leader)**
JAYA PRIYA R	**511319104027**
JEFFONIA A	**511319104029**
ARTHI J	**511319104006**

In partial fulfilment of the award of degree

Of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KINGSTON ENGINEERING COLEGE,VELLORE

ANNA UNIVERSITY: CHENNAI 600 025

Executed By:

IBM

ABSTRACT:

In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personalized job suggestions. Although in the literature exists a variety of techniques and strategies used as part of job recommender systems, most of them fail to recommending job vacancies that fit properly to the job seekers profiles. Thus, the contributions of this work are threefold, we:

i) made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites;
ii) put forward the proposal of a framework for job recommendation based on professional skills of job seekers;
iii) carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. We thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue.

This article presents a recommender system that aims to help job seekers to find suitable jobs. First, job offers are collected from job search websites then they are prepared to extract meaningful attributes such as job titles and technical skills. Job offers with common features are grouped into clusters. As job seeker like one job belonging to a cluster, he will probably find other jobs in that cluster that he will like as well. A list of top n recommendations is suggested after matching data from job clusters and job seeker behavior, which consists on user interactions such as applications, likes and rating.

CHAPTER NO	TITLE	PAGE
1	**INTRODUCTION** 1.1 Project Overview 1.2 Purpose	
2	**LITERATURESURVEY** 2.1 Existing problem 2.2 References 2.3 Problem Statement Definition	
3	**IDEATION & PROPOSED SOLUTION** 3.1 Empathy Map Canvas 3.2 Ideation& Brainstorming 3.3 Proposed Solution 3.4 Problem Solution fit	
4	**REQUIREMENTANALYSIS** 4.1 Functional requirement 4.2 Non-Functional requirements	
5	**PROJECT DESIGN** 5.1 Data Flow Diagrams 5.2 Solution & Technical Architecture 5.3 User Stories	

CHAPTER NO	TITLE	PAGE
6	**PROJECT PLANNING& SCHEDULING** 6.1 Sprint Planning & Estimation 6.2 Sprint Delivery Schedule 6.3 Reports from JIRA	
7	**CODING & SOLUTIONING** 7.1 Feature 1 7.2 Feature 2 7.3 Database Schema	
8	**TESTING** 8.1 Test Cases 8.2 User Acceptance Testing	
9	**RESULTS** 9.1 Performance Metrics	
10	**ADVANTAGES&DISADVANTAGES**	
11	**CONCLUSION**	
12	**FUTURE SCOPE**	

CHAPTER NO	TITLE	PAGE
13	**APPENDIX** 13.1 SourceCode 13.2 GitHub& Project DemoLink	

1 INTRODUCTION:

When the whole world is coming back on its feet, those businesses affected by this pandemic disease slowly tries to gain back the momentum it lost. Now is the time when the companies or businesses seek to invest in human resources, which would help them to gain the momentum it lost during this period. When the governments across the world ask businesses to halt the operation in the effort of controlling the pandemic, many companies asked their employees to work remotely. In contrast, many other companies started to reduce their operational cost by terminating employees who were in permanent and contract roles.

Individuals who lost their job to the consequence of shutdown are a waiting for their next opportunity. Naturally, we human tries to strive through all difficulties to serve the purpose of our life. A daily job provides a sense of purpose to an individua l(stillman, 2019), and he tries to get better at it, which results in leaving current employment and looking for a new one; this is a constant cycle of the hiring process.

To serve the constant cycle of the hiring process in the job applicant's perspective, many job companies have come up with

solutions for providing the job board. Here a seeker looks up for the job he would find relevant to him and apply for it. As there are many job boards, applicants tend to use the tool that provides better services to them, services such as writing a CV, creating a job profile, and recommending new jobs to a job seeker.

1.1 Project overview:

Job applicants have become more persistent and proactive in searching for new opportunities that fit their skills. However, companies that are targeting these job seekers are finding it challenging to identify the job seeker's skill and provide personalized job recommendations

1.2 Purpose:

- It is an approach to an information retrieval or machine learning problem. The assumption made in content-based filtering is that user prefers item with similar properties. Content-based filtering recommends items to the user whose properties are similar to the item which the user has previously shown interest.
- As powerful data filtering tools, recommendation systems use algorithms and data analysis techniques to recommend the most relevant product/items to a particular user. The main aim of any recommendation engine is to stimulate demand and actively engage users.
- What is it's Final Goal? A Recommender System predicts the possibility that a user would favor an item. Based on former user interaction with the data source that the system takes the learning from (besides the data from other users, or historical trends), the system is capable of recommending an item to a user.

1.3 Summary of literature Survey

A review has been made on different approaches in various research articles. A detailed study is made by referring various papers of different fields, such as load demand forecasting, data mining techniques, soft computing approaches and different application that users time series process. The methods involved each article is discussed briefly, it also includes merits and demerits of each work. Finally a summarise is made based on the survey.

1.3.1 A Career Path Recommendation Framework :

In today's world, recommendation systems are used to the problem are in connection overload in many areas allowing users to focus on important information based on their interests. One of the areas where such systems can play a major role is in helping students achieve their career goals by generating personalized job and skill recommendation. At present, there are many job posting websites providing a huge amount of a information and students need to spend hours to find job that match their interests. At the same time excisting job recommendation systems only consider the user’s field to interest, but do not take into consideration the user’s profile and skill, which can generate more relevant career recommendation for users. This article was published in March 2017 and authors of this article are: Bharat patel ; Varun Kakuste ; Magadalini

1.4 Existing Problem:

With continuing advances in artificial intelligence (AI), recommendations are no longer aimed at general audiences, or even those of a certain segment. Using deep learning-based recommendation engines, marketers today can target consumers with hyper-personalized recommendations at the individual level, based on metrics like persona, location, interests, real-time online behavior and so on. This will not only allow marketers to drive online traffic through retargeting ads or email marketing, but also reduce customer irritation and churn rates.

Personalized product recommendations also help engage customers by serving them with products or services that are highly relevant to them. This will encourage higher average order values and increase conversions. In the long term, using personalized recommendations will show your customers that you understand and value them, increasing customer satisfaction and loyalty

Reference:

1. Slamet C., Andrian R., Maylawati D. S. A., Darmalaksana W., Ramdhani M. A. Web scraping and Naïve Bayes classification for job search engine. IOP Conference Series: Materials Science and Engineering . 2018;288(1) doi: 10.1088/1757-899X/288/1/012038.012038 [CrossRef] [Google Scholar]

2. Musale D. V., Nagpure M. K., Patil K. S., Sayyed R. F. Job recommendation system using profile matching and web crawling. Int. J. Adv. Sci. Res. Eng. Trends. . 2016;1 [Google Scholar]

3. Kethavarapu U. P. K., Saraswathi S. Concept based dynamic ontology creation for job recommendation system. Procedia Computer Science . 2016;85:915–921. doi: 10.1016/j.procs.2016.05.282. [CrossRef] [Google Scholar]

4. Parihar A. S., Gupta Y. K., Singodia Y., Singh V., Singh K. A comparative study of image dehazing algorithms. Proceedings of the 2020, 5th International Conference on Communication and Electronics Systems (ICCES); 2020, June; Coimbatore, India. pp. 766–771. [CrossRef] [Google Scholar]

5. Mishra R., Rathi S. ICDSMLA 2019 . Singapore: Springer; 2020. Efficient and scalable job recommender system using collaborative filtering; pp. 842–856. [Google Scholar]

6. Mhamdi D., Moulouki R., El Ghoumari M. Y., Azzouazi M., Moussaid L. Job recommendation based on job profile clustering and job seeker behavior. Procedia Computer Science . 2020;175:695–699. doi: 10.1016/j.procs.2020.07.102. [CrossRef] [Google Scholar]

7. Desai V., Bahl D., Vibhandik S., Fatma I. Implementation of an automated job recommendation system based on candidate profiles. Int. Res. J. Eng. Technol . 2017;4(5):1018–1021. [Google Scholar]

8. Indira V., Rathika S. International Research Journal of Engineering and Technology (IRJET) 4. Vol. 7. ISSN: ISSN: 2395-0056; 2020. A study on E-recruitment and it's present condition towards job seekers; pp. 3753–3758. [Google Scholar]

9. Pradhan R., Varshney J., Goyal K., Kumari L. International Conference on Innovative Computing and Communications . New York, NY, USA: Springer Nature; 2022. Job recommendation system using content and collaborative-based filtering; pp. 575–583. [CrossRef] [Google Scholar]

10. Belsare R. G., Deshmukh V. M. Employment recommendation system using matching, collaborative filtering and content- based recommendation. International Journal of Computer Applications Technology and Research . 2018;7(6):215–220. doi: 10.7753/IJCATR0706.1003. [CrossRef] [Google Scholar]

11. Manjare P., Kumbhar J., Ovhal S., Munde R. An effective job recruitment system using content-based filtering. International Journal of Engineering & Technology . 2017;4(3):2395–0056. [Google Scholar]

12. Akshaya C. P. Enhancement of recommender system using collaborative filtering. International Research Journal of Engineering and Technology . 2018;5(4):2198–2200. [Google Scholar]

13. Jain H., Kakkar M. Job recommendation system based on machine learning and data mining techniques using RESTful API and android IDE. Proceedings of the 2019, 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence); 2019, January; Noida, India. pp. 416–421. [CrossRef] [Google Scholar]

14. Yi P., Yang C., Li C., Zhang Y. A job recommendation method optimized by position descriptions and resume information.

Proceedings of the 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC); 2016, October; Xi'an, China. pp. 761–764. [CrossRef] [Google Scholar]

15. Apaza H., Rubin de Celis Vidal A. A., Chire Saire J. E. Future of Information and Communication Conference . New York, NY, USA: Springer; 2021. Job recommendation based on curriculum vitae using text mining; pp. 1051–1059. [CrossRef] [Google Scholar]

16. Nasution D., Sitorus Z. Enhance web-based job search recommendation system of hybrid-based recommendation. Budapest International Research and Critics Institute (BIRCI-Journal): Humanities and Social Sciences . 2021;4(3):7214–7221. [Google Scholar]

17. Zhou Q., Liao F., Ge L., Sun J. Personalized preference collaborative filtering: job recommendation for graduates. Proceedings of the 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation; 2019, August; Leicester, UK. pp. 1055–1062. [CrossRef] [Google Scholar]

18. Patel R., Vishwakarma S. K. ICT Systems and Sustainability: Proceedings of ICT4SD . Vol. 1. New York, NY, USA: Springer Nature; 2020. An efficient approach for job recommendation system based on collaborative filtering; p. p. 169. [Google Scholar]

19. Chou Y. C., Yu H. Y. Based on the application of AI technology in resume analysis and job recommendation. Proceedings of the 2020, IEEE International Conference on Computational Electromagnetics (ICCEM); 2020, August; Singapore. pp. 291–296. [CrossRef] [Google Scholar]

20. Punitavathi D., Shinu V., Kumar S., Sp V. P. Online job and candidate recommendation system. International Research Journal of

Multidisciplinary Technovation . 2019;1:84–89. doi: 10.34256/irjmt19212. [CrossRef] [Google Scholar]

21. Yadalam T. V., Gowda V. M., Kumar V. S., Girish D., Namratha M. Career recommendation systems using content based filtering. Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES); 2020, June; Coimbatore, India. pp. 660–665. [CrossRef] [Google Scholar]

22. Almalis N., Tsihrintzis G., Karagiannis N. A new content-based recommendation algorithm for job recruiting. Proceedings of the International Conference on Innovative Techniques and Applications of Artificial Intelligence; 2015, December; Corfu, Greece. Springer; pp. 393–398. [CrossRef] [Google Scholar]

23. Suharyadi J., Kusnadi A. Stay at home reservation: the mitigation step in covid-19 pandemic. International Journal of New Media Technology . 2018;5(2):116–120. [Google Scholar]

24. Rimitha S. R., Abburu V., Kiranmai A., Chandrasekaran K. Ontologies to model user profiles in personalized job recommendation. Proceedings of the 2018 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER); 2018, August; Mangalore, India. pp. 98–103. [CrossRef] [Google Scholar]

25. Shamrat F. J. M., Tasnim Z., Rahman A. S., Nobel N. I., Hossain S. A. An effective implementation of web crawling technology to retrieve data from the world wide web (WWW) International Journal of Scientific & Technology Research . 2020;9(01):1252–1256. [Google Scholar]

26. Nigam H., Biswas P. Applications of Artificial Intelligence and Machine Learning . Singapore: Springer; 2021. From web scraping to web crawling; pp. 97–112. [Google Scholar]

27. Karthikeyan T., Sekaran K., Ranjith D., Balajee J. M. Personalized content extraction and text classification using effective

web scraping techniques. International Journal of Web Portals . 2019;11(2):41–52. [Google Scholar]

28. Kumar N., Aggarwal D. LEARNING-based focused WEB crawler. IETE Journal of Research . 2021:1–9. [Google Scholar]

29. Rai D., Kumar N., Mor J. Review on improving performance of web crawler and search system Architecture. International Journal of Advanced Studies of Scientific Research . 2018;3(10) [Google Scholar]

30. Kamishima T., Akaho S., Asoh H., Sakuma J. Recommendation independence. Proceedings of the Conference on Fairness, Accountability and Transparency 187-201); 2018, January; New York, NY, USA. [Google Scholar]

31. Kumar M., Bindal A., Gautam R., Bhatia R. Keyword query based focused Web crawler. Procedia Computer Science . 2018;125:584–590. [Google Scholar]

32. Diouf R., Sarr E. N., Sall O., Birregah B., Bousso M., Mbaye S. N. Web scraping: state-of-the-art and areas of application. Proceedings of the 2019, IEEE International Conference on Big Data (Big Data); 2019, December; Los Angeles, CA, USA. pp. 6040–6042. [CrossRef] [Google Scholar]

33. Glez-Peña D., Lourenço A., López-Fernández H., Reboiro-Jato M., Fdez-Riverola F. Web scraping technologies in an API world. Briefings in Bioinformatics . 2014;15(5):788–797. [PubMed] [Google Scholar]

34. Uzun E. A novel web scraping approach using the additional information obtained from web pages. IEEE Access . 2020;8 doi: 10.1109/ACCESS.2020.2984503.61726 [CrossRef] [Google Scholar]

35. Gupta M., Kumar N., Singh B. K., Gupta N. NSGA-III-Based deep-learning model for biomedical search engine. Mathematical Problems in Engineering . 2021;2021:8. doi: 10.1155/2021/9935862.9935862 [CrossRef] [Google Scholar]

36. Mor J., Kumar N., Rai D. An improved crawler based on efficient ranking algorithm. International Journal of Advanced Trends in Computer Science and Engineering . 2019;8:119–125. doi: 10.30534/ijatcse/2019/04822019. [CrossRef] [Google Scholar]

37. Mor J., Rai D., Kumar N. An XML based web crawler with page revisit policy and updation in local repository of search engine. International Journal of Engineering & Technology . 2018;7(3):1119–1123. doi: 10.14419/ijet.v7i3.12924. [CrossRef] [Google Scholar]

38. Mor J., Kumar N., Rai D. Research on mechanism and challenges in meta search engines. International Journal of Innovative Technology and Exploring Engineering . 2019;8(9):281–284. [Google Scholar]

39. Kumar N., Nath R., Kherwa P. An automated framework based on TLS to choose best search engine in a particular domain. International Journal of Computer Application . 2013;83(14):42–48. doi: 10.5120/14520-2920. [CrossRef] [Google Scholar]

40. Mor J., Kumar N., Rai D. Effective presentation of results using ranking & clustering in meta search engine, COMPUSOFT. An International Journal of Advanced Computer Technology . 2018;7(12) [Google Scholar]

41. Kumar N., Dahiya P. Weighted similarity page rank: an improvement in WPR and WSR. International Journal of Computer Engineering and Applications . 2017;11(VIII):1–11. [Google Scholar]

42. Kumar N., Nath R. A meta search engine approach for organizing web search results using ranking and clustering. International Journal of Computer . 2013 [Google Scholar]

43. Kumar N., Tyagi D., Awasthi S. Survey on crawling techniques. Proceedings of the 5th International Conference on “Computing for Sustainable Global Development; March, 2018; New Delhi, India. pp. 2449–2455. [Google Scholar]

44. Kumar N. Document clustering approach for meta search engine. IOP Conference Series: Materials Science and Engineering . 2017;225

doi: 10.1088/1757-899X/225/1/012291.012291 [CrossRef] [Google Scholar]

45. Kumar N., Tyagi D., Awasthi S., Mor J. Change detection of web page in focused crawling system. International Journal of Control Theory and Applications . 2017;10(6):671–676. [Google Scholar]

46. Kumar N., Singh P. Meta search engine with semantic analysis and query processing. International Journal of Computational Intelligence Research ISSN . 2017;13(8):0973–1873. [Google Scholar]

47. Kumar N. Segmentation based twitter opinion mining using ensemble learning. International Journal on Future Revolution in Computer Science & Communication Engineering . 2017;3(9) [Google Scholar]

48. Nath R., Kumar N., Tuteja S. Intelligent Distributed Computing . New York, NY, USA: Springer; 2015. A survey on reduction of load on the network; pp. 239–249. [Google Scholar]

49. Kumar N., Das N. N., Gupta D., Gupta K., Bindra J. Efficient automated disease diagnosis using machine learning models. Journal of Healthcare Engineering . 2021;2021:13. doi: 10.1155/2021/9983652.9983652 [PMC free article] [PubMed] [CrossRef] [Google Scholar]

50. Singrodia V., Mitra A., Paul S. A review on web scrapping and its applications. Proceedings of the 2019, International Conference on Computer Communication and Informatics; 2019, January; Coimbatore, India. pp. 1–6. [CrossRef] [Google Scholar]

51. Nath R., Kumar N. A novel parallel domain focused crawler for reduction in load on the network. International Journal of Computational Engineering Research . 2012;2(7):77–84. [Google Scholar]

52. Shrivastava V. A methodical study of web crawler. Journal of Engineering Research and Application . 2018;8(11):1–8. [Google Scholar]

53. Amudha S., Phil M. Web crawler for mining web data. International Research Journal of Engineering and Technology (IRJET) . 2017;4(2):2395–0072. [Google Scholar]

54. Nassar N., Jafar A., Rahhal Y. A novel deep multi-criteria collaborative filtering model for recommendation system. Knowledge-Based Systems . 2020;187 doi: 10.1016/j.knosys.2019.06.019.104811 [CrossRef] [Google Scholar]

55. Fu M., Qu H., Yi Z., Lu L., Liu Y. A novel deep learning-based collaborative filtering model for recommendation system. IEEE Transactions on Cybernetics . 2018;49(3):1084–1096. doi: 10.1109/TCYB.2018.2795041. [PubMed] [CrossRef] [Google Scholar]

2.3 Problem Statement Definition

The current era is fully evolved with computational technologies with the latest and significant trends in Business Analytics in making an efficient business decision. Also, the advent of the internet changes our day-to-day life as we utilize computational technologies and intelligence for not only our business but also to make life more convenient. It is possible to make life easy going through computer solutions and the internet, this blog throws lights on how computer solutions in the terms of Recommendation System improve daily life activities that are directly related to business productivity and hence left us unanswerable.

3. IDEATION & PROPOSED SOLUTION

A product recommendation engine is essentially a solution that allows marketers to offer their customers relevant product recommendations in real-time. As powerful data filtering tools, recommendation systems use algorithms and data analysis techniques to recommend the most relevant product/items to a particular user.

Recommendations should be one-sentence, succinct, and start with an action verb (create, establish, fund, facilitate, coordinate, etc.). They should use a “SMART” format (Specific, Measurable, Attainable, Realistic, Timely). Each recommendation should be followed by a few sentences of explanatory text.Choose strong, accurate adjectives over general terms or cliches. Think about what the person reading the letter might want to know. Include details or insight that wouldn't necessarily show up on a resume or application. Your letter should help the recipient get to know the person you are recommending.

Assess your ability in each skill as accurately as you can. Ask yourself if you have used this skill a little or a lot. For each skill, write a sentence showing how you've used that skill. Then write a sentence showing how you could use that skill in the job you would like.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An empathy map canvas is a more in-depth version of the original empathy map, which helps identify and describe the user’s needs and pain points.

And this is valuable information for improving the user experience.

Teams rely on user insights to map out what is important to their target audience, what influences them, and how they present themselves. This information is then used to create personas that help teams visualize users and empathize with them as individuals, rather than just as a vague marketing demographic or account number.

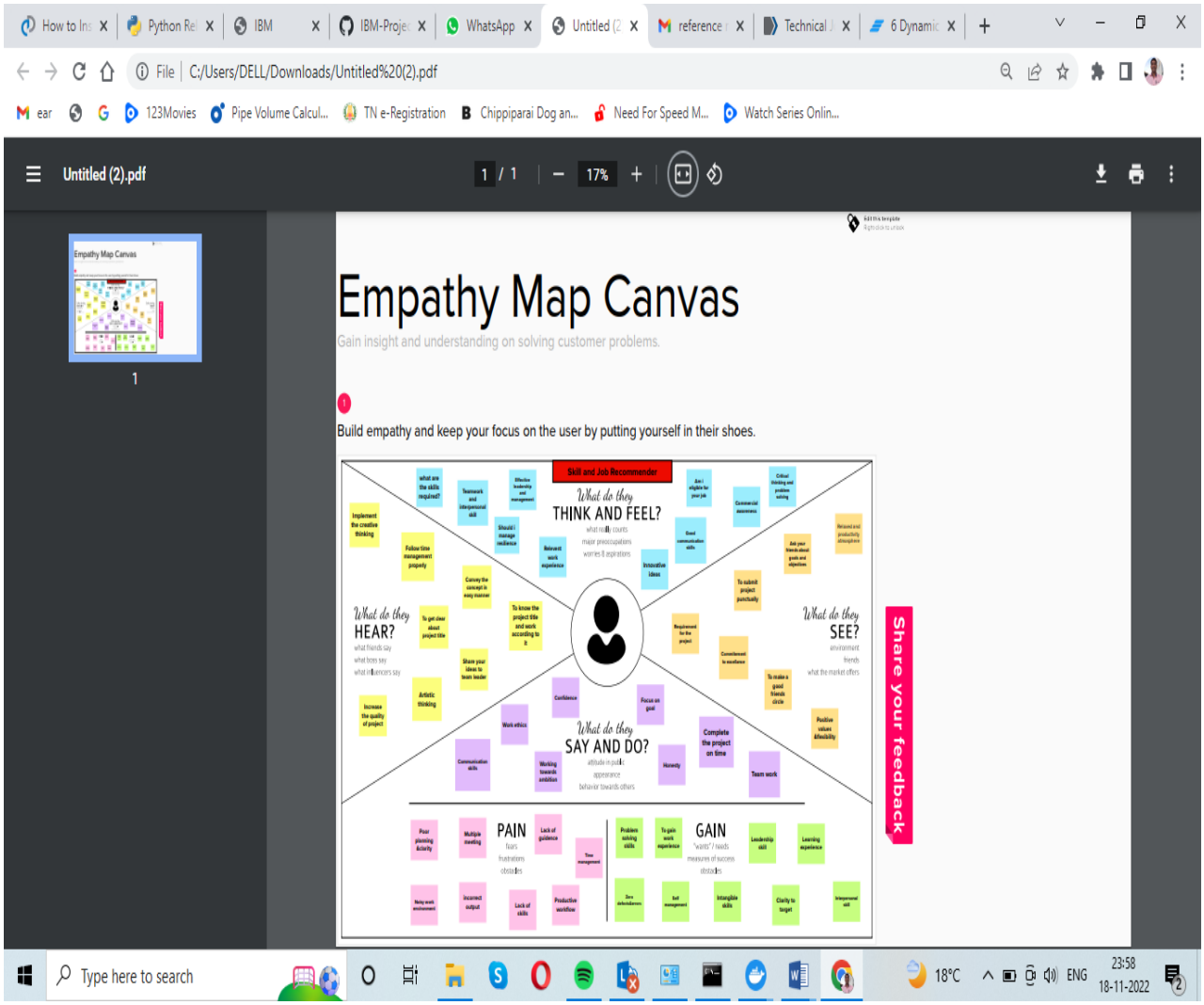
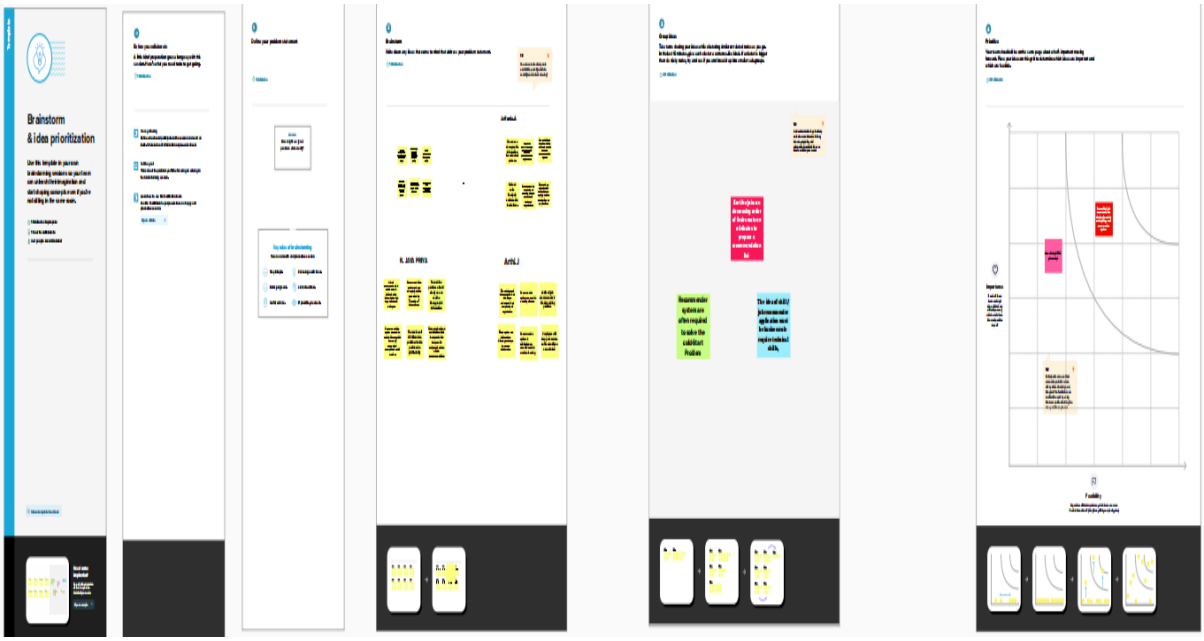
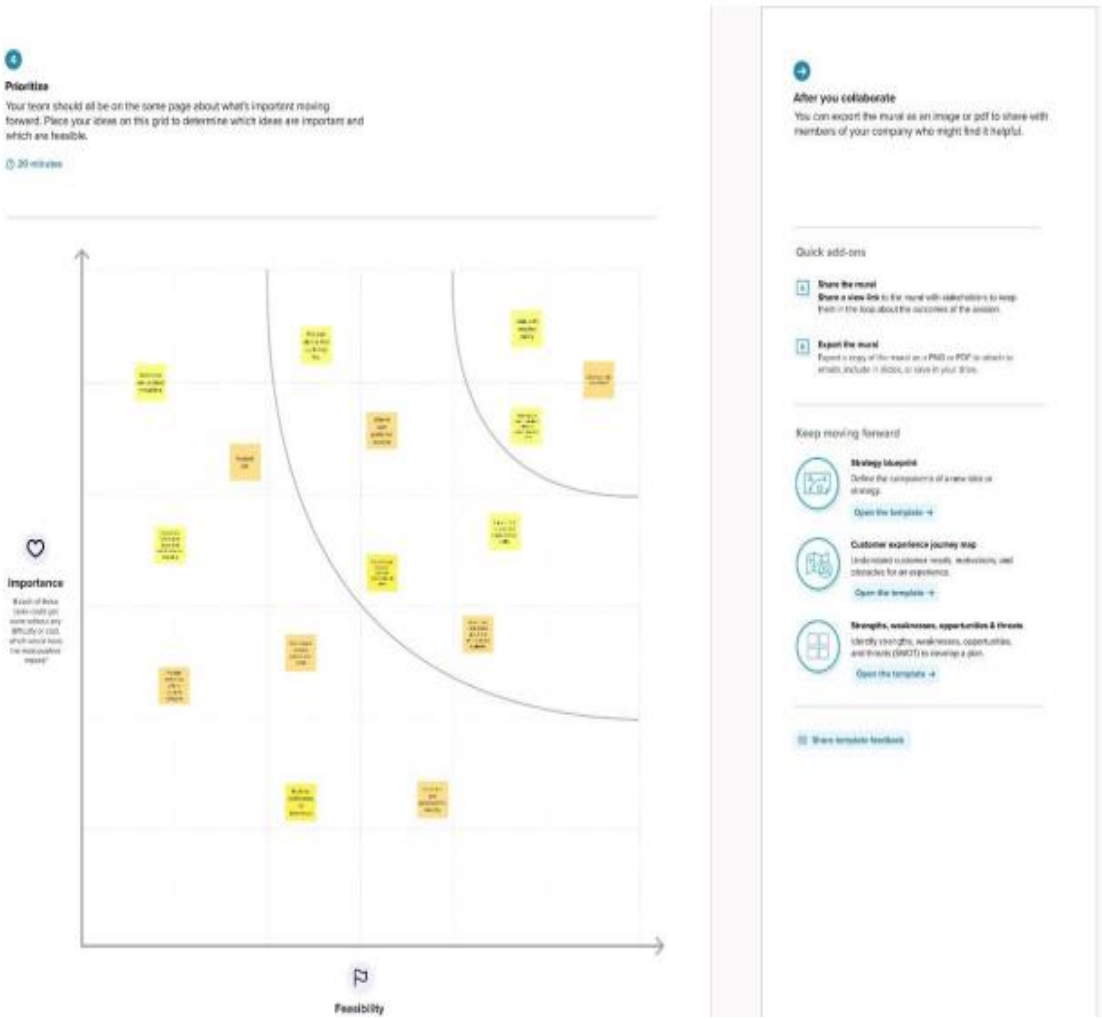


Fig: Empathy Map Canvas





(3.2 IDEATION & BRAINSTORMING)

3.3 Proposed Solution

S.no	Parameters	Description
1.	Problem Statement (Problem to be solved)	Having lots of skills but wondering which job will best suit you? Don't need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job. To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.
2.	Idea/solution	The contributions of this work are threefold, we: i) made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites ii) put forward the proposal of a framework for job recommendation based on professional skills of job seekers iii) carried out an evaluation to quantify empiíically the recommendation abilities of two state-oftheart methods, considering different configurations, within the proposed fssramework. We thus present a general panorama of job recommendation

		task aiming to facilitate research and real-world application design regarding this important issue
3.	Novelty / Uniqueness	The best position are suggested to any person according to her skills.While the position of known profiles are assumed PNT2022TMID39871 should be noted that there are usually multiple advisable positions corresponding to a set of skills.A recommendation system should return a set of most likely positions and all of them can be equally valid. The recommendation method we use is simply based on representing both positions and profiles as comparable vectors and seeking for each profile the positions with the most similar vectors.
4.	Business Model (Revenue Model)	We can provide the application for job seekers in a subscription based and we can share the profiles with companies and generate the revenue by providing them best profiles.
5.	Social Impact / Customer Satisfaction	Students will be benefited as they will get to know which job suits them based on their skill set and therefore Lack of Unemployment can be reduced
6.	Scalability of the Solution	Data can be scaled up and scaled down according to number of current job openings available.

3.4 Problem Solution fit

Problem-Solution fit canvas **Skill / Job Recommender Application**

Define CS, fit into CC

1.CUSTOMER SEGMENT(S)

- Job Seeker
- Job Recommender

6. CUSTOMER CONSTRAINTS

- Lack of awareness about a job Openings.
- Personal data security.
- Vulnerable to employment scams

5. AVAILABLE SOLUTIONS

- Linked in, indeed, and Naukri are some of the leading sources for job opportunities.
- They intimate user (Job seeker) with a notification about a recent Job Openings based on their skillset.
- Premium user will get more features including learning resources, etc..

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2.JOBS-TO-BE-DONE / PROBLEMS

Job Seeker:

- Finding desired job is not an easy task.
- They need to gain knowledge before applying a particular job.
- They should Be aware of fraudulent job post.

Job Recruiter:

- They need to find a skilled candidate for her company.
- The hiring process takes so much time to complete.
- Filtering candidates is difficult.

9. PROBLEM ROOT CAUSE

- Increasing in population as well as increasing in graduates on particular domain leads to Job Crisis.
- The education system does not fulfil and focus on individual person skill development.

7.BEHAVIOUR

- Learn and see more about a Job Openings in job posting website.
- Develop and improve her knowledge.
- Connect with recruiters on Linked in platform and maintain a friendly connection with people.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3.TRIGGERS

- Financial Problem
- Societal pressure
- Dissatisfaction of Job
- Finds a better way to improve her knowledge as well as career growth.

4.EMOTIONS:

BEFORE

- Sad, depressed, and low confidence.
- Fear of Rejection before attending any hiring process.

AFTER

- Highly Motivated
- Gained confidence to do any task.

10. YOUR SOLUTION

- A Fake Job Offer is detected and removed automatically.
- Recommend a skill to job seeker for a particular Job Openings.
- A notification will be Send via email regarding job openings.
- Learning resources will be provided, then it will improve the user knowledge and skills.

8. CHANNELS of BEHAVIOUR

ONLINE:

- Apply and maintain a connection with recruiters.
- Also search about job openings.

OFFLINE

- Learn and gain the required skills in open Source platform as well as in our Job Website.

Extract online & offline CH of BE

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution

FR.NO	Functional Requirement	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mail
FR-2	User Confirmation	Confirmation via Mail Confirmation via OTP

4.2 Non-Functional requirements

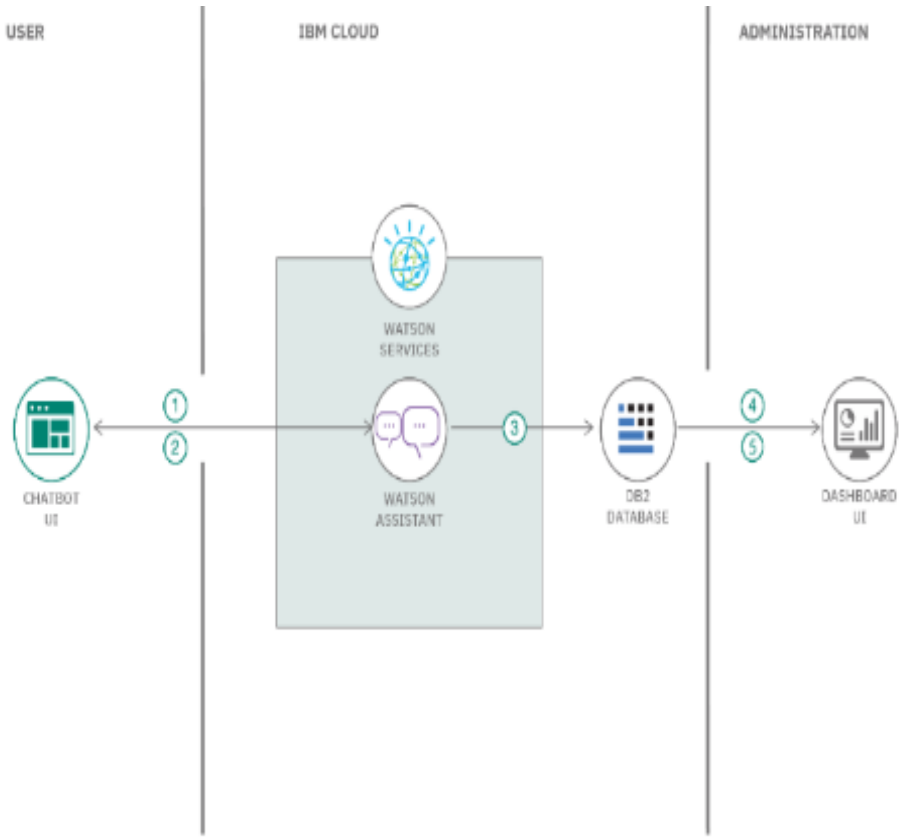
Following are the non-functional requirements of the proposed solution

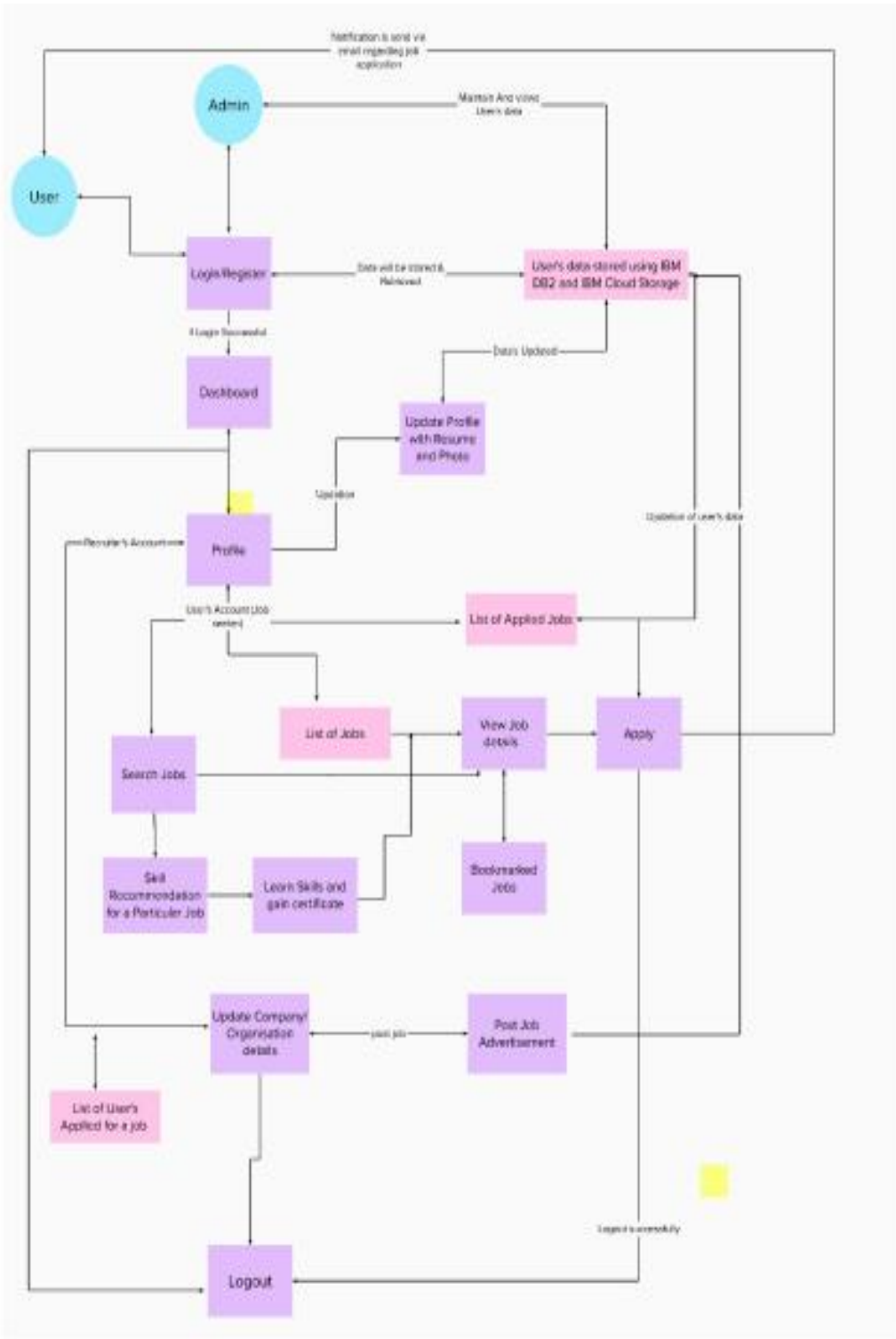
FR No.	Non-Functional Requirement	Description
FR-1	Usability	The candidates who use the system should be students or skilled graduates
FR-2	Security	The users can secure their profile using steganography. The security on their databases may include firewalls to prevent unauthorized access.
FR-3	Reliability	Applicants can access their resume 98% of the time without failure.
FR-4	Performance	The websites load time should not be more than one second.

FR-5	Availability	Employers can post jobs on the website throughout the weeks at any time during the week
FR-6	Scalability	It is the ability to appropriately handle increasing Workloads without performance degradation or its ability to quickly enlarge

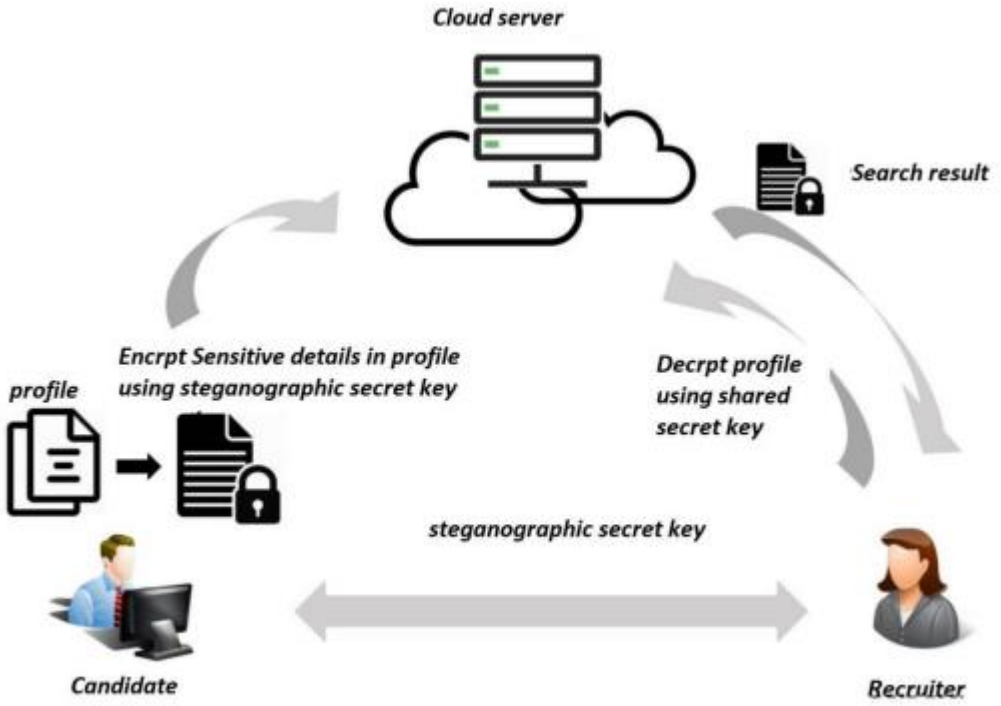
5. PROJECT DESIGN:

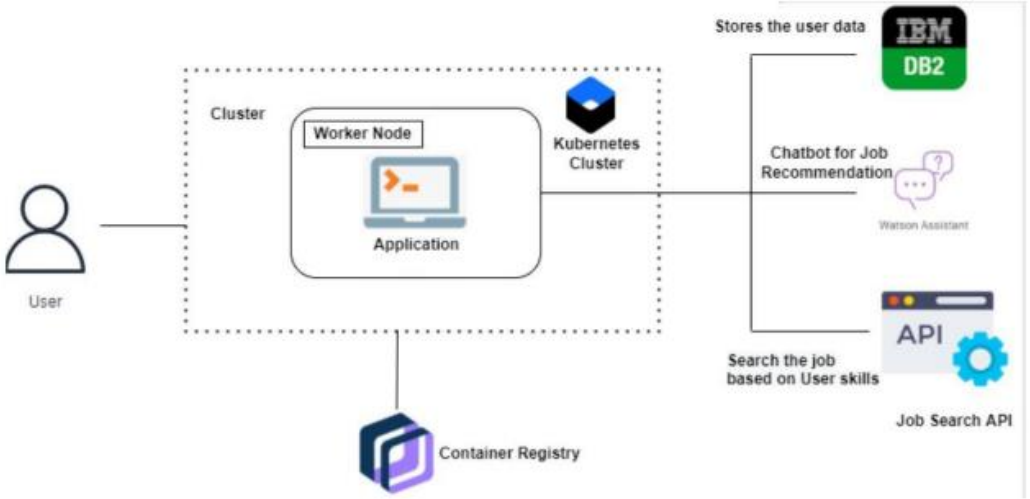
5.1 Data Flow Diagrams




Admin
User
Login/Register
User's data stored using IBM DB2 and IBM Cloud Storage
Dashboard
Update Profile with Resume and Photo
Profile
List of Applied Jobs
List of Jobs
View Job details
Apply
Search Jobs
Skill Recommendation for a Particular Job
Learn Skills and gain certificate
Bookmarked Jobs
Update Company/ Organization details
Post Job Advertisement
List of User's Applied for a job
Logout

5.2 Solution & Technology Architecture





5.3 User Stories

Use the below template to list all the user stories for the product

User Type	Functional Requirement(Epic)	User Story Number	User Story/Task	Acceptance criteria	Prior-ity	Rel e-ase
Customer(Mobile User)	Registration	USN -1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN -2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN -3	As a user, I can register for the application through online websites	I can register & access the dashboard with online website Login	Low	Sprint-2
		USN -4	As a user, I can register	I can receive	Medium	Sprint-1

			for the application through Gmail	confirmation on Gmail & click confirm		
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard					
Customer(Web user)		USN-6	As a user, I can able to take up the skill assessment and view the appropriate test score. Based on the skill sets I can able to get personalised job recommendations.	I can receive job recommendations	High	Sprint-1
Customer Care Executive		USN-7	As a customer care executive, we provide 24/7 chatbot suppor	24/7 chatbot support	High	Sprint-1

Adminstration		USN-8	As an administrator, I can able to view the progress and make required changes in the project	Deploy user specific and personalised job recommendations	High	Sprint-1

6. PROJECT PLANNING& SCHEDULING

6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User story Number	User story/task	Story Points	Priori-ty	Team Members
Sprint-1	Candidate	USN-1	The candidate can searches for their desired jobs based on their owned and acquired skills.	20	high	Kanimozhi.A Jaya Priya.R Jeffonia.A Arthi.J
Sprint-2	Job providers	USN-2	The role of the job providers is to searches for the deserved candidate with the knowledge and skill required for their companies.	20	high	Kanimozhi.A Jaya Priya.R Jeffonia.A Arthi.J
Sprint-3	Chatbot	USN-3	The users can directly talk with the chatbot regarding the availability of candidates and jobs.	20	high	Kanimozhi.A Jaya Priya.R Jeffonia.A Arthi.J
Sprint-4	Final delivery	USN-4	Container of the application using docor,Kubernetes and deployment of the application	20	high	Kanimozhi.A Jaya Priya.R Jeffonia.A Arthi.J

Project Tracker, Velocity & Burndown Chart:

Sprint	Total Story points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story End Date(as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	26 Oct	31 Oct	20	31 Oct
Sprint-2	20	6 Days	02 Nov	07 Nov	20	07 Nov
Sprint-3	20	6 Days	09 Nov	14 Nov	20	14 Nov
Sprint-4	20	6 Days	15 Nov	20 Nov	20	20 Nov

7. CODING & SOLUTIONING

TEMPLATE:

7.1 Index.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Nilesh Hadalgi</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.mi
n.css" rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohh
puuCOmLASjC" crossorigin="anonymous">
</head>

<body>
    <nav class="navbar fixed-top navbar-dark bg-primary d-flex justify-
content-center">
        <a class="navbar-brand" href="#!">Skill Set Based
Recommandation System</a>
    </nav>

    <div class="container" style="margin-top: 100px;">
        <div class="row">
            <div style="margin-top: 20px;" class="col-lg-6">
                <h1 style="padding: .375rem .375rem;">Enter Your
Requirements</h1>
                <form action = "http://localhost:5000/" method = "POST">
```

```
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="html">HTML</label>
                    <input name="html" type="text" class="form-control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="python">Python</label>
                    <input name="python" type="text" class="form-
control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="java">Java</label>
                    <input name="java" type="text" class="form-control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="c">C</label>
                    <input name="c" type="text" class="form-control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="javascript">JavaScript</label>
                    <input name="javascript" type="text" class="form-
control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="candidate">Number of Candidate</label>
                    <input name="candidate" type="text" class="form-
control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <button style="padding: .375rem .375rem;"
type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>
        </div>
        <div style="margin-top: 20px;" class="col-lg-6">
            <table class="table">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Name</th>
```

```
                    <th>Score</th>
                </tr>
            </thead>
            <tbody>
            {% for key,value in result %}
                <tr>
                    <th class="nr" scope="row">1</th>
                    <td>{{value}}</td>
                    <td>{{key}}</td>
                </tr>
            {% endfor %}

            </tbody>
        </table>
    </div>
</div>
<div style="margin-top: 20px;" class="alert alert-danger"
role="alert">
    <strong>Note: </strong>
    <br>
    *Enter the requirements rated on scale of 5 and are true intiger.
    <br>
    *This model uses Euclidian Distance score to recommand
user/groups.
    <br>
    *Rest other features.
</div>
</div>
</body>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bund
le.min.js"
integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+
JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
<script type="text/javascript">
var a = document.getElementsByClassName("nr");
for (var i = 0; i < a.length; i++) {
    a[i].innerHTML = (i+1)+".";
}
</script>

</html>
```

7.2 Readme.md

Euclidian Distance

In mathematics, the Euclidean distance between two points in Euclidean space is the length of a line segment between the two points. It can be calculated from the Cartesian coordinates of the points using the Pythagorean theorem, therefore occasionally being called the Pythagorean distance.

Formula

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

p, q = two points in Euclidean n-space

q_i, p_i = Euclidean vectors, starting from the origin of the space (initial point)

n = n-space

7.3 TechStack

Python3.x
Flask
HTML
CSS
JavaScript
CSV & JSON (data-set Files)

7.4 Main.py

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Nilesh Hadalgi</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.mi
n.css" rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohh
puuCOmLASjC" crossorigin="anonymous">
```

```
</head>

<body>
   <nav class="navbar fixed-top navbar-dark bg-primary d-flex justify-
content-center">
      <a class="navbar-brand" href="#!">Skill Set Based
Recommandation System</a>
   </nav>

   <div class="container" style="margin-top: 100px;">
      <div class="row">
         <div style="margin-top: 20px;" class="col-lg-6">
            <h1 style="padding: .375rem .375rem;">Enter Your
Requirements</h1>
            <form action = "http://localhost:5000/" method = "POST">
               <div style="padding: .375rem .375rem;" class="form-
group">
                  <label for="html">HTML</label>
                  <input name="html" type="text" class="form-control">
               </div>
               <div style="padding: .375rem .375rem;" class="form-
group">
                  <label for="python">Python</label>
                  <input name="python" type="text" class="form-
control">
               </div>
               <div style="padding: .375rem .375rem;" class="form-
group">
                  <label for="java">Java</label>
                  <input name="java" type="text" class="form-control">
               </div>
               <div style="padding: .375rem .375rem;" class="form-
group">
                  <label for="c">C</label>
                  <input name="c" type="text" class="form-control">
               </div>
               <div style="padding: .375rem .375rem;" class="form-
group">
                  <label for="javascript">JavaScript</label>
                  <input name="javascript" type="text" class="form-
control">
               </div>
               <div style="padding: .375rem .375rem;" class="form-
group">
                  <label for="candidate">Number of Candidate</label>
                  <input name="candidate" type="text" class="form-
```

```
control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <button style="padding: .375rem .375rem;"
type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>
        </div>
        <div style="margin-top: 20px;" class="col-lg-6">
            <table class="table">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Name</th>
                        <th>Score</th>
                    </tr>
                </thead>
                <tbody>
                {% for key,value in result %}
                    <tr>
                        <th class="nr" scope="row">1</th>
                        <td>{{value}}</td>
                        <td>{{key}}</td>
                    </tr>
                {% endfor %}

                </tbody>
            </table>
        </div>
    </div>
    <div style="margin-top: 20px;" class="alert alert-danger"
role="alert">
        <strong>Note: </strong>
        <br>
        *Enter the requirements rated on scale of 5 and are true intiger.
        <br>
        *This model uses Euclidian Distance score to recommand
user/groups.
        <br>
        *Rest other features.
    </div>
  </div>
</body>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bund
```

```
le.min.js"
    integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+
JcXn/tWtIaxVXM"
    crossorigin="anonymous"></script>
<script type="text/javascript">
    var a = document.getElementsByClassName("nr");
    for (var i = 0; i < a.length; i++) {
        a[i].innerHTML = (i+1)+".";
    }
</script>

</html>
```

Python3.x
Flask
HTML
CSS
JavaScript
CSV & JSON (data-set Files)

```
from flask import Flask, render_template, request
import recommender

app = Flask(_name_)

@app.route('/', methods=['POST', 'GET'])
def hello_world():
    if request.method == 'POST':
        result = request.form
        requirement = {"REQUIREMENT": {
            "HTML": int(result['html']),
            "Python": int(result['python']),
            "Java": int(result['java']),
            "C": int(result['c']),
            "JavaScript": int(result['javascript'])}}
        num_of_candidate = int(result['candidate'])
        result = recommender.topMatches(requirement,
recommender.dataFrame, "REQUIREMENT", num_of_candidate)
        print(result)
        return render_template("index.html", result=result)

    return render_template("index.html", result=[("name","Score")])


if _name_ == '_main_':
```

```
    app.run(debug=True)
```

7.4 Recommended.py

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Nilesh Hadalgi</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.mi
n.css" rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohh
puuCOmLASjC" crossorigin="anonymous">
</head>

<body>
    <nav class="navbar fixed-top navbar-dark bg-primary d-flex justify-
content-center">
        <a class="navbar-brand" href="#!">Skill Set Based
Recommandation System</a>
    </nav>

    <div class="container" style="margin-top: 100px;">
        <div class="row">
            <div style="margin-top: 20px;" class="col-lg-6">
                <h1 style="padding: .375rem .375rem;">Enter Your
Requirements</h1>
                <form action = "http://localhost:5000/" method = "POST">
                    <div style="padding: .375rem .375rem;" class="form-
group">
                        <label for="html">HTML</label>
                        <input name="html" type="text" class="form-control">
                    </div>
                    <div style="padding: .375rem .375rem;" class="form-
group">
                        <label for="python">Python</label>
                        <input name="python" type="text" class="form-
control">
                    </div>
                    <div style="padding: .375rem .375rem;" class="form-
```

```
group">
                    <label for="java">Java</label>
                    <input name="java" type="text" class="form-control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="c">C</label>
                    <input name="c" type="text" class="form-control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="javascript">JavaScript</label>
                    <input name="javascript" type="text" class="form-
control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <label for="candidate">Number of Candidate</label>
                    <input name="candidate" type="text" class="form-
control">
                </div>
                <div style="padding: .375rem .375rem;" class="form-
group">
                    <button style="padding: .375rem .375rem;"
type="submit" class="btn btn-primary">Submit</button>
                </div>
            </form>
        </div>
        <div style="margin-top: 20px;" class="col-lg-6">
            <table class="table">
                <thead>
                    <tr>
                        <th>#</th>
                        <th>Name</th>
                        <th>Score</th>
                    </tr>
                </thead>
                <tbody>
                {% for key,value in result %}
                    <tr>
                        <th class="nr" scope="row">1</th>
                        <td>{{value}}</td>
                        <td>{{key}}</td>
                    </tr>
                {% endfor %}
```

```
                </tbody>
            </table>
        </div>
    </div>
    <div style="margin-top: 20px;" class="alert alert-danger"
role="alert">
        <strong>Note: </strong>
        <br>
        *Enter the requirements rated on scale of 5 and are true intiger.
        <br>
        *This model uses Euclidian Distance score to recommand
user/groups.
        <br>
        *Rest other features.
    </div>
  </div>
</body>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bund
le.min.js"
  integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+
JcXn/tWtIaxVXM"
  crossorigin="anonymous"></script>
<script type="text/javascript">
  var a = document.getElementsByClassName("nr");
  for (var i = 0; i < a.length; i++) {
    a[i].innerHTML = (i+1)+".";
  }
</script>

</html>
```

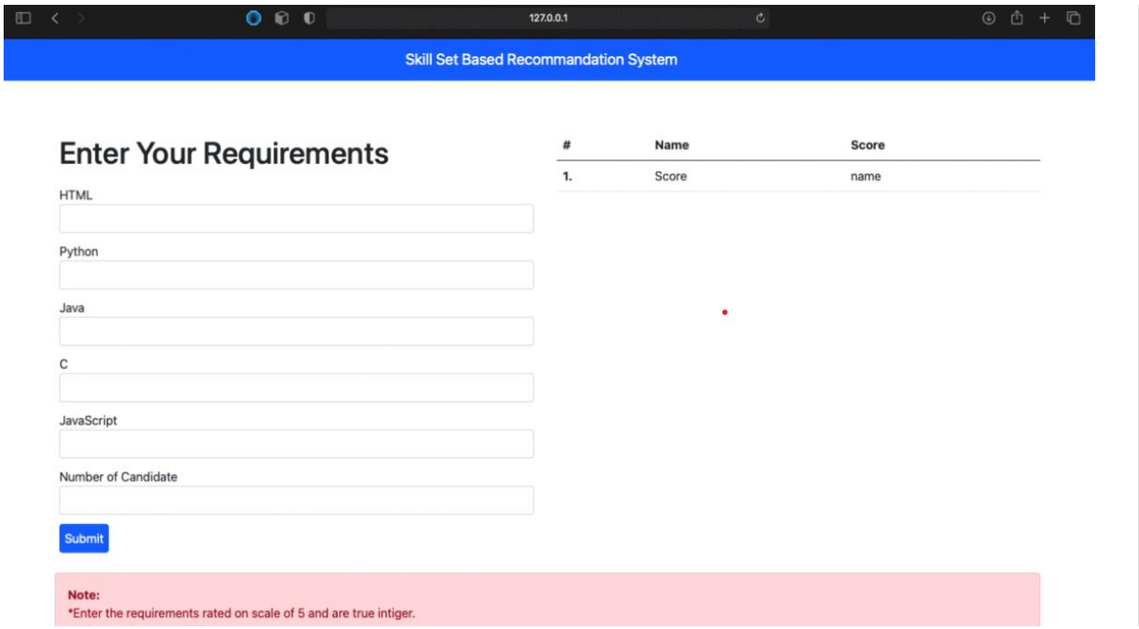
Python3.x
Flask
HTML
CSS
JavaScript
CSV & JSON (data-set Files)

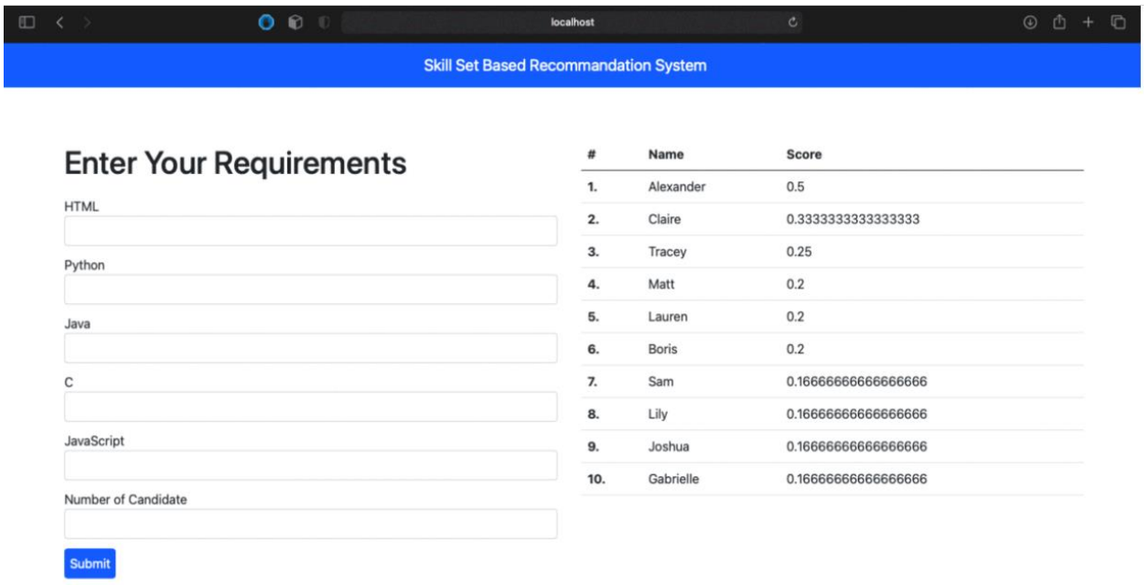
```
from flask import Flask, render_template, request
import recommender

app = Flask(_name_)

@app.route('/', methods=['POST', 'GET'])
def hello_world():
  if request.method == 'POST':
```

```
        result = request.form
        requirement = {"REQUIREMENT": {
            "HTML": int(result['html']),
            "Python": int(result['python']),
            "Java": int(result['java']),
            "C": int(result['c']),
            "JavaScript": int(result['javascript'])}}
        num_of_candidate = int(result['candidate'])
        result = recommender.topMatches(requirement,
recommender.dataFrame, "REQUIREMENT", num_of_candidate)
        print(result)
        return render_template("index.html", result=result)

    return render_template("index.html", result=[("name","Score")])


if _name_ == '_main_':
    app.run(debug=True)
import json
from math import sqrt

dataFrame = json.load(open("1000DATASET.json"))


def sim_distance(reqirements_json, prefs, person1, person2):
    si = {}
    for item in reqirements_json[person1]:
        if item in prefs[person2]:
            si[item] = 1
    if len(si) == 0: return 0

    sum_of_squares = sum(
        [pow(reqirements_json[person1][item] - prefs[person2][item], 2)
for item in reqirements_json[person1] if
         item in prefs[person2]])
    return 1 / (1 + sum_of_squares)


def topMatches(reqirements_json, prefs, person, n,
similarity=sim_distance):
    scores = [(similarity(reqirements_json, prefs, person, other), other) for
other in prefs if other != person]
    scores.sort()
    scores.reverse()
    print(scores)
```

return scores[0:n]

8.TESTING AND RESULTS





9. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

While such platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the opportunity of recruiting. The recommender system technology aims to help users in finding items that

match their personnel interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently

DISADVANTAGES:

disadvantages are the cold start, scalability, and low behavior. Its process starts with cleaning and building the database and obtaining data attributes. Then, the cosine similarity function is used to find the correlation between the previous user and the available list.

10. CONCLUSION

we used a literature analysis of manyjjournals and proceedings related to the recruiting process and the job recommendation researches. We have seen from our literature review and from the challenges that faced the holistic e-recruiting platforms, an increased need for enhancing the quality of candidates/job matching.The recommender system technologiesaccomplished significant success in a broad range ofapplications and potentially a powerful searching and recommending techniques. Consequently, there is a great opportunity for applying these technologies inrecruitment environment to improve the matching quality.This survey shows that several approaches for job recommendation have been proposed, and many techniques combined in order to produce the best fit between jobs and candidates.

We presented state of theart of job recommendation as well as, a comparativestudy for its approaches that proposed by literatures. Additionally,we reviewed typical recommender system techniques and the recruiting process related issues. We conclude that the field of job recommendations is still unripe and require further improvements.As part of our ongoing research, we aim to build a new recommendation approach and test with real data for employee and staffing data from large companies

11. FUTURE SCOPE

IS technologies for human resource management in general and recruiting processes in particular. Most companies put the focus on their own e-recruiting platforms as primary recruitment channels. Job ads are published automatically on the job portal as soon as they are entered into the system. On the other hand, the applicant creates a profile to apply it for one of the listed job positions. The user profile is stored in the system, letting the applicant reuse it for other job position. The last functionality gives the companies possibility to create the applicants pool. Thus, the companies achieved a uniform view for all applicants‟ data in one candidate pool. This pool is used by the recruitment department to find the applicant documents. Appropriate applicants‟documents are directed to the human resource departments for more processing. In addition, the system supports all required communication processes as well as tracks applicant status inside the application process (Malinowski et al., 2005). The e-recruiting platforms are usually based on Boolean search and filtering techniques that cannot sufficiently capture the complexity of a person-job fit as selection decisions (Malinowski et al., 2006). Manyliteratures have been applied the recommender system concept into the job problem. Malinowski et al. (2008) determined that, we must consider unary attributes such as individual skills, mental abilities and personality that control the fit between the individual and the tasks to be accomplished, as well as the relational attributes that determine the fit between the individual and the upcoming team members. In this context literatures usually distinguish between (1) person-job, (2) person-team and (3)

person-organization fits (Sekiguchi, 2004). Thus, the recruitment approach must cover all this aspects. Keim (2007) argues that transferring recommender system approach to search for persons is a challenging but promising goal. Therefore, many recommendation approaches applied for matching candidates and jobs to overcome the previous challenges of holistic e-recruiting platforms (Laumer and Eckhardt, 2009). It is been a decade new uptades and technologies are developing this will show an immense impact on society for umemployment and utilising the skilful peoples globally