SKILL/JOB RECOMMENDER APPLICATION

A PROJECT REPORT BY TEAM PNT2022TMID28874

Team Lead :

Cathlene Hena J (411719104008)

Team Members :

- Kaviya R (411719104026)
- Kaviya Varshini S (411719104027)
- Saranya G (411719104044).

Industry Mentor(s) Name :

Krishna Chaitanya

Faculty Mentor(s) Name :

AISHWARYA

Category :

Cloud App Development

SKILL/JOB RECOMMENDER APPLICATION

INDEX

1.INTRODUCTION

1.1 PROJECT OVERVIEW

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.

1.2 PURPOSE

The main idea is to match jobseeker’s skills to job. It is also to classify certain group of jobs' candidates according to the feature requirements and recommend jobs that match their features. The recommender systems are being used to figure out the interested items for a certain user by employing a variety of information resources that is related to the users and items. All young aspirants have lot of skills, but are not aware of the numerous fields that are having great demand. There is an indispensable need to make people aware of the vast job fields available. Hence making people match their skill to job. This platform serves as a bridge, helping aspirants choose their passionate job and thereby love their profession.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

There are various disadvantages such as less secure, can build native apps only, etc that overcomes using the proposed system.

- Less secure

The concerns regarding security in Job recommender apps still persists due to several reasons. First major cause is due to its open source property, which means that the possible code vulnerabilities become common knowledge after they have been found.The second reason regarding its security is the low entry barrier for novice programmers. As a result, a number of websites and apps are developed by inexperienced coders, trainees, or even hobbyists. The shoddy results of their work contribute to the rumors and facts regarding overall bad security and performance.

- Data Storage

The memory consumption and data overhead incase of existing systems are very high. For memory intensive tasks such as online shopping rather than preferring languages like php, python, etc web stacks can be preferred. As Structured Query Language (sql) is being used, storing unstructured and semistructured data is quite a tedious process. As the efficiency of an e-commerce app relies on the efficient management of databases, these traditional existing apps lack this property.

- Longer time

The time taken to build a full stack e-commerce application is way more large when compared while creating with web stacks such as MERN, MEAN, etc. The various phases involved in creating a web app needs to be properly planned and executed. Our proposed solution will enhance the time taken in building and releasing the webapp. Releasing the app without any defects is another major goal, hence properly planned building and timely delivery are being implemented by the proposed system

- Native apps

Native apps are built for a specific operating system. A native app developed for the iOS operating system won't work on Android devices and vice- versa. If an app is developed for iOS, it will remain exclusive to that operating system. If at all the app has to support the Android version, a new app has to be built again for the Android operating system. Software's used to develop native apps generally would be Objective-C or Swift for iOS, Java and ADT for Android operating system and NET(C#) for Windows operating system.

2.2 REFERENCES

[AlO12] Shaha T Al-Otaibi and Mourad Ykhlef. "A survey of job recommender systems". In: International

Journal of the Physical Sciences 7.29 (2012), pp. 5127–5142. issn: 19921950. doi: 10.5897/IJPS12.

482.

[Den15] N Deniz, A Noyan, and O G Ertosun. "Linking Person-job Fit to Job Stress: The Mediating Effect of

Perceived Person-organization Fit". In: Procedia - Social and Behavioral

Sciences 207 (2015), pp. 369–

376.

[Dia13] M Diaby, E Viennet, and T Launay. “Toward the next generation of recruitment tools: An online

social network-based job recommender system”. In: Proc. of the 2013 IEEE/ACM Int. Conf. on

Advances in Social Networks Analysis and Mining, ASONAM 2013 (2013), pp. 821–828. doi: 10.

1145/2492517.2500266.

[Dia14] M Diaby and E Viennet. “Taxonomy-based job recommender systems on Facebook and LinkedIn

profiles”. In: Proc. of Int. Conf. on Research Challenges in Information Science (2014), pp. 1–6. issn:

21511357. doi: 10.1109/RCIS.2014.6861048.

[Kus15] M Kusner et al. “From word embeddings to document distances”. In: Proc. of the 32nd Int. Conf. on

Machine Learning, ICML’15. 2015, pp. 957–966.

[Mik13a] T Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”.

In: Proc. of the 26th Int. Conf. on Neural Information Processing Systems - Volume 2. NIPS’13.

Lake Tahoe, Nevada, 2013, pp. 3111–3119. url: http://dl.acm.org/citation.cfm?id=2999792.

2999959.

[Mik13b] T Mikolov et al. “Efficient estimation of word representations in vector space”. In: arXiv preprint

arXiv:1301.3781 (2013).

[Sal88] G Salton and C Buckley. “Term-weighting approaches in automatic text retrieval”. In: Information

Processing and Management 24.5 (1988), pp. 513–523. issn: 0306-4573. doi: https://doi.org/10.

1016/0306- 4573(88)90021- 0. url: http://www.sciencedirect.com/science/article/pii/

0306457388900210.

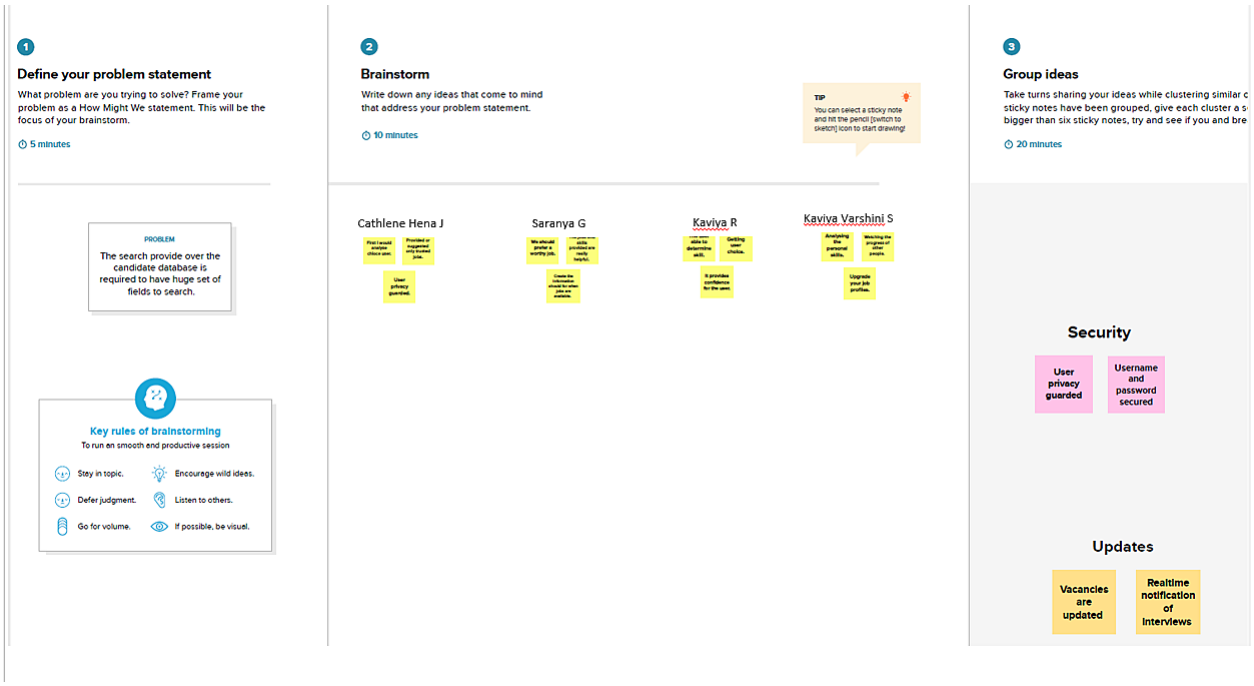
2.3 PROBLEM STATEMENT DEFINITION

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. 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

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION AND BRAINSTORMING

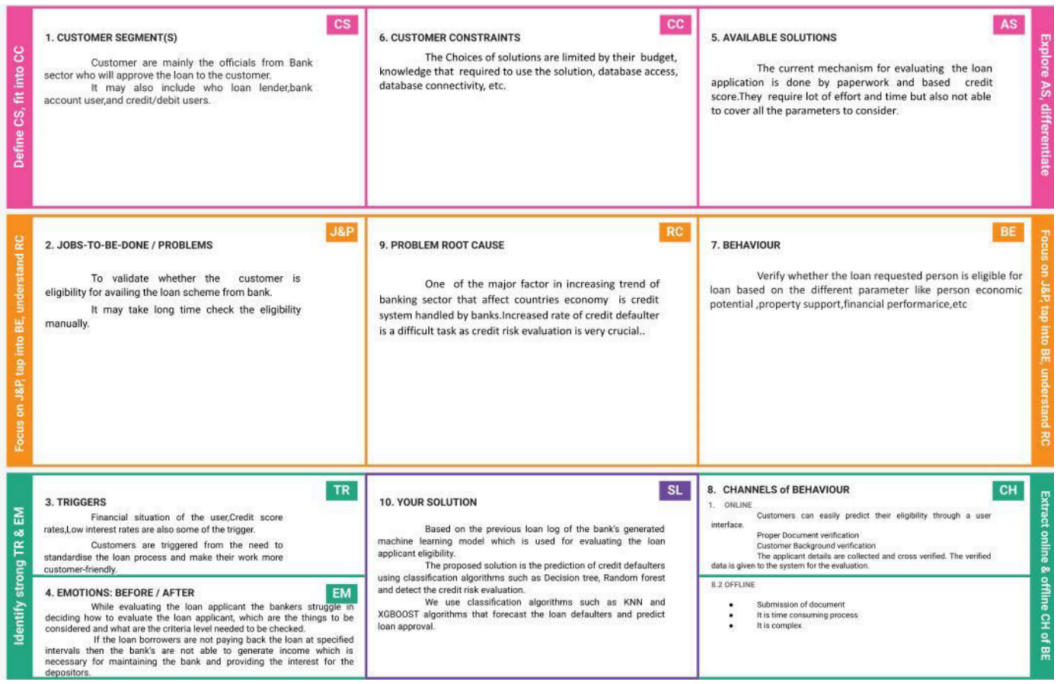


3.3 PROPOSED SOLUTION

S.No.	Parameter	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 description	The solution proposed is that a new data set will be formed based on the skill sets of the job seekers and the set of job vacancies. Thus by matching the skills set of people to the requirements, job seekers are recommended jobs. Thus aspirant's skills are matched to their preferred domains, thereby exploring various job fields. Even chatbot assistance is provided.

4.	Social Impact / Customer Satisfaction	This helps aspirants explore the unexplored fields which actually has demand. Every skills are matched to their corresponding requirements. The companies in here will be verified so job seekers feel safe and secured.
5.	Business Model (Revenue Model)	We can share the details of the job seekers to companies with the permission of users, thereby generate revenue. We can even allocate certain honours to subscribed users, resulting in income.
6.	Scalability of the Solution	Requests per second can be handled by load balancers. Any number of users can be handled, though vast number of users simultaneously request access to data.

3.4 PROPOSED SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

Following are the functional requirements of the proposed solution

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Sign in / Login	Register with username, password
FR-2	Profile Registration	Register with username, password, email, qualification, skills. This data will be stored in a database.
FR-3	Job profile display	Display job profiles based on availability, location, skills.
FR-4	Chatbot	A chat on the webpage to solve user queries and issues
FR-5	Job Registration	The company's registration/Description details will be sent to the registered email id of the user.
FR-6	Logout	Use logout option after completing job registration process

4.2 NON FUNCTIONAL REQUIREMENT

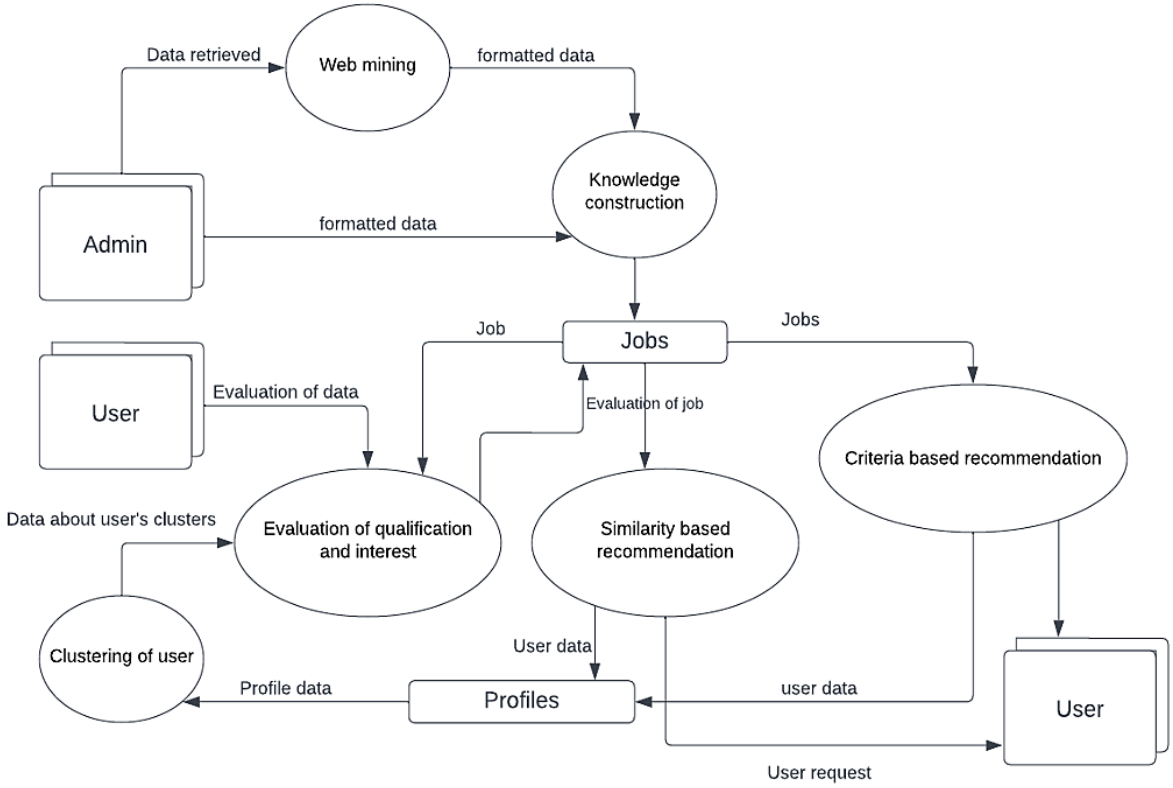
Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description

NFR-1	Usability	The webpage will be designed in such a way that any non-technical user can easily navigate through it and complete the job registration work. (easy and simple design)
NFR-2	Security	Using of python flask to cloud connect will provide security to the project. Database will be safely stored in DB2
NFR-3	Reliability	To make sure the webpage doesn't go down due to network traffic
NFR-4	Performance	Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic
NFR-5	Availability	The webpage will be available to all users (network connectivity is necessary) at any given point of time.
NFR-6	Scalability	Increasing the storage space of database can increase the number of users. Add some features in future to make the webpage unique and attractive.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service

4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service
8.	External API-1	Purpose of External API used in the application	Job search API, IBM registry
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, Docker.

Application Characteristics

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Python-Flask

2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	IAM Controls
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	IBM DB2
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	IBM Cloud Load Balancers

User Type	5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Auto Scaling on Cloud server	**Priority**	**Release**

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 Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1

	Dashboard	USN-5	As a user, I can access my dashboard after signing in.	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Access	USN-6	As a user, I can setup a profile, and basic details by signing in.			
		USN-7	As a user, I will upload my resume, certificates, and other requirements.	I can perform several task in the application	Medium	Sprint-1
Customer Care Executive	Chatbot	USN-8	As a user, I can seek guidance from the customer care executive.		High	Sprint-1
Administrator	DBMS	USN-9	As a administrator, I can keep the applications of your organization relies on running.	I can perform various modifications in the applications.	High	Sprint-1

5.3 USER STORIES

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	UI Creation, Creating Registration page, Login page.	10	Medium	CATHLEN HENA J KAVIYA VARSHINI S KAVIYA R SARANYA G
Sprint-2	Chatbot Development	USN-4	Building a chatbot.	10	High	CATHLENE HENA SARANYA G
Sprint-3	Integration and Containerization	USN-5	Integrating chatbot to the HTML page and containerizing the app.	20	Medium	CATHLEN HENA J KAVIYA VARSHINI S KAVIYA R SARANYA G
Sprint-4	Upload Image and deployment	USN-6	Upload the image to the IBM Registry and deploy it in the Kubernetes Cluster.	20	High	KAVIYA R SARANYA G

6.PROJECT PLANNING & SCHEDULING

a.SPRINT PLANNING & ESTIMATION

Project Tracker, Velocity & Burndown Chart

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022		
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022		
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022		

Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Burndown Chart:

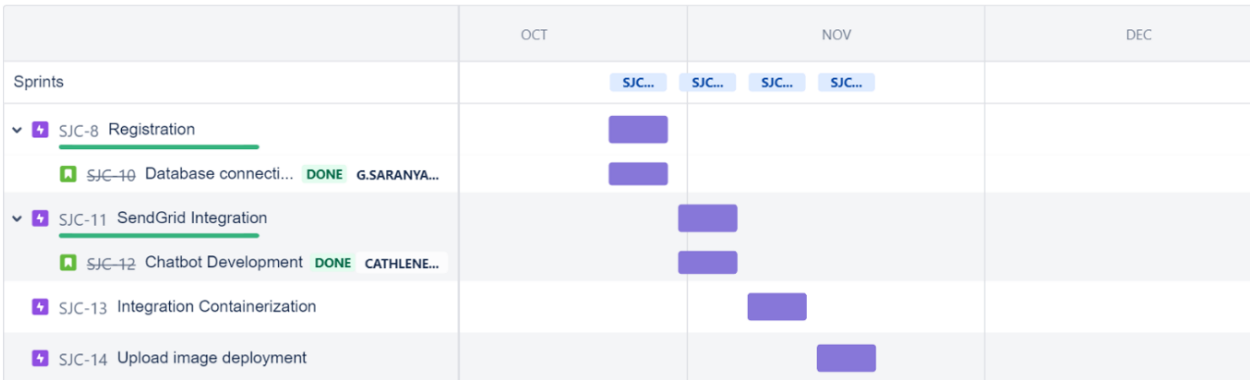
A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing

measurable progress over time.

a. **SPRINT DELIVERY PLAN**

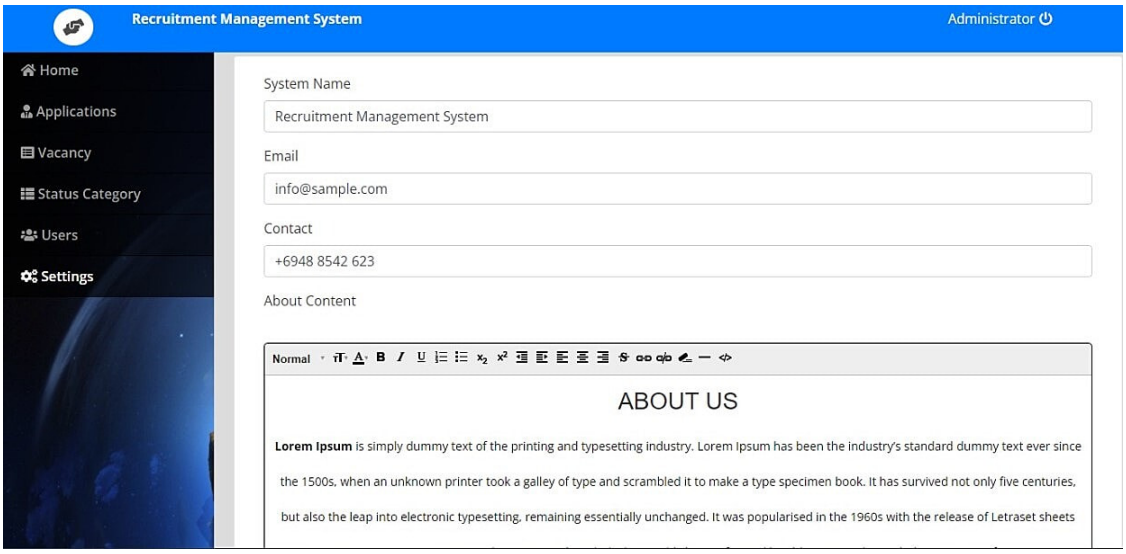
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	UI Creation, Creating Registration page, Login page.	10	Medium	CATHLEI HENA J KAVIYA VARSHINI S KAVIYA R SARANYA G
Sprint-1	Database Connectivity	USN-2	Viewing Products and Connecting UI with Database.	10	High	KAVIYA R CATHLENE HENA J
Sprint-2	SendGrid Integration	USN-3	Send Grid Integration with python code.	10	Low	CATHLENE HENA J KAVIYA VARSHINI S
Sprint-2	Chatbot Development	USN-4	Building a chatbot.	10	High	CATHLENE HENA SARANYA G
Sprint-4	Upload Image and deployment	USN-6	Upload the image to the IBM Registry and deploy it in the Kubernetes Cluster.	20	High	KAVIYA R SARANYA G

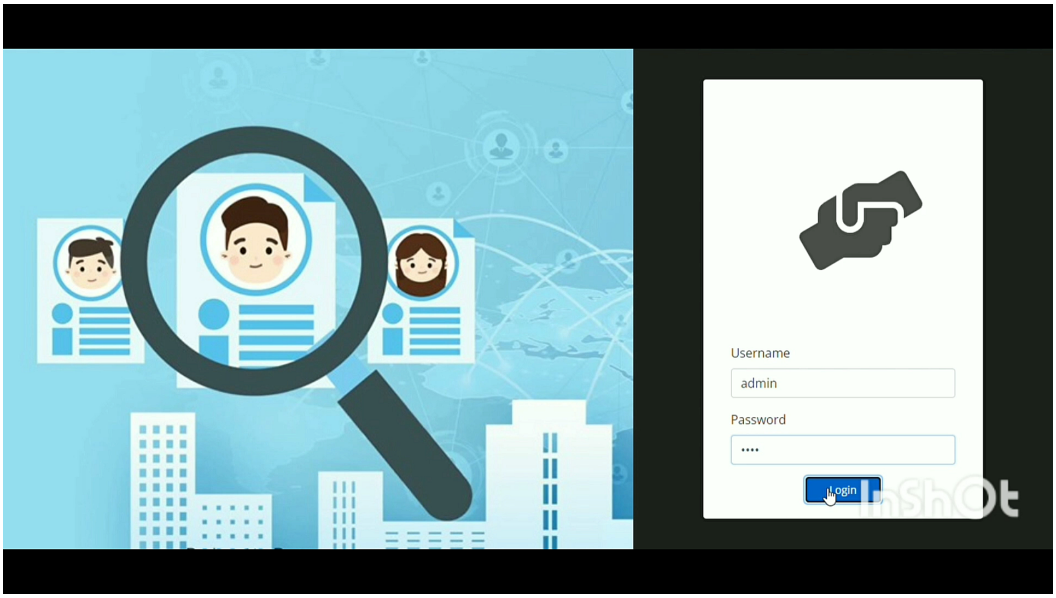
6.3 REPORTS FROM JIRA



7.CODING & SOLUTIONING

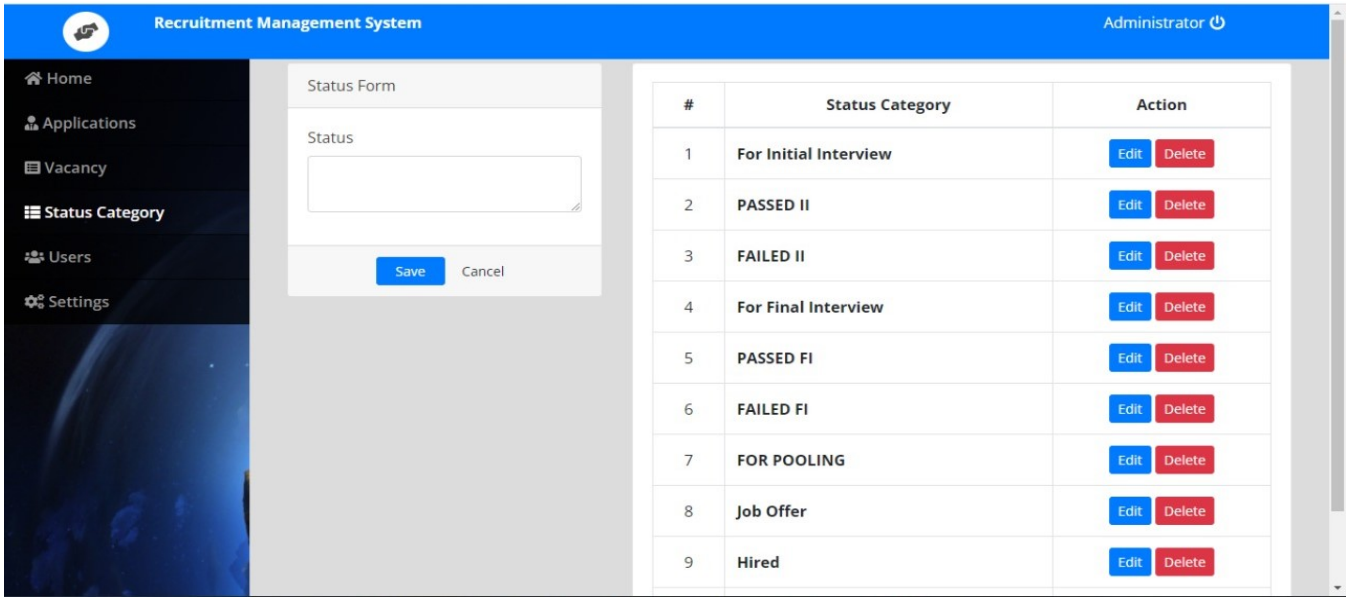
1. FEATURE 1- Register & Login

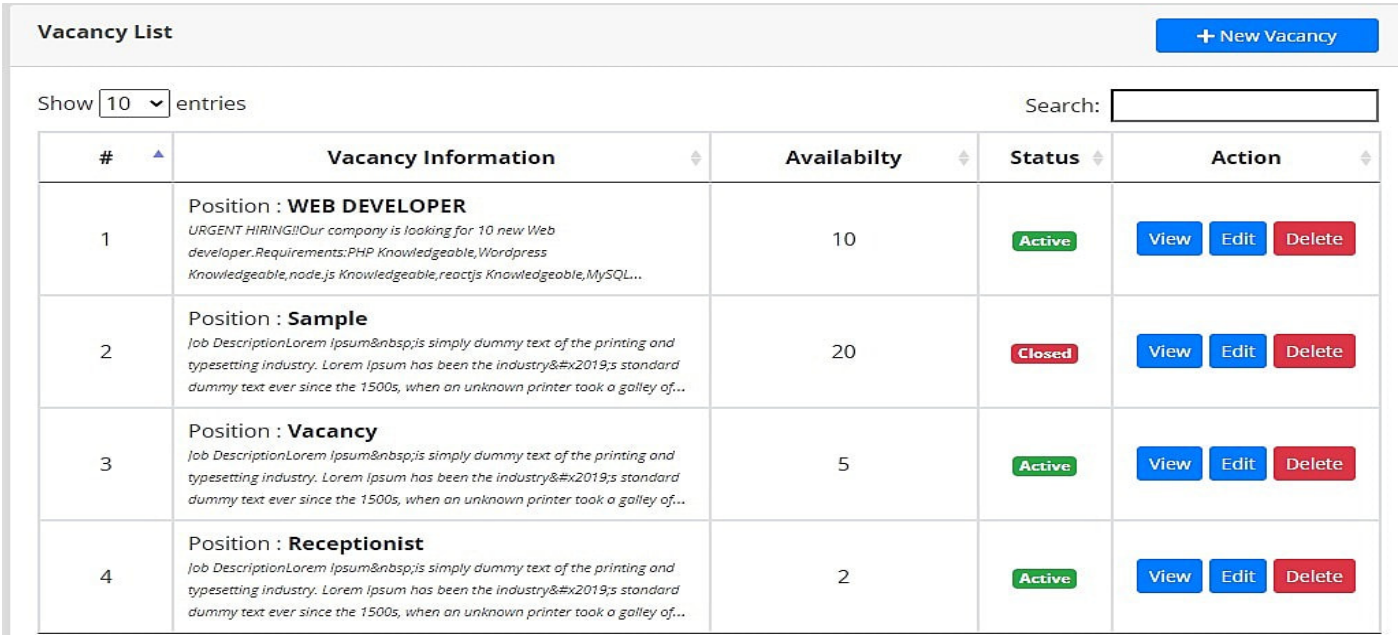




Jobseekers can register using their credentials, which will be saved in IBM DB2 cloud. So, whenever the user wants to visit the application he/she just can login using the saved credentials.

2. FEATURE 2: Admin & Status





Another added feature is that the status of each applicant can be accessed by the admin, that status will be sent to the jobseeker through their respective mail. The users can check their application status using their login.

8. TESTING

8.1 TEST CASES

Test case ID	Feature Type	Component	Test Scenario
LoginPage_TC_OO1	Functional	Home Page	Verify user is able to see the Login/Signup popup when user clicked on My account button
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials
LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials

LoginPage_TC_OO4	Functional	Login page	Verify user is able to log into application with InValid credentials

LoginPage_TC_OO5	Functional	Login page	Verify user is able to log into application with InValid credentials

Pre-Requisite	**Steps To Execute**	**Test Data**
	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup displayed or not	index.html

	1.Enter URL and click go 2.Click on My Account dropdown button 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account link e.Last password? Recovery password link	index.html
	1.Enter URL(index.html) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: saranyaganesan156@gmail.com password: Testing123

	1.Enter URL(index.html) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter valid password in password text box 5.Click on login button	Username: cathy56hena@gmail password: Testing123
	1.Enter URL(index.html) and click go 2.Click on My Account dropdown button 3.Enter Valid username/email in Email text box 4.Enter Invalid password in password text box 5.Click on login button	Username: varskaav589@gmail.com password: Testing1236786867868768 76

	1.Enter URL(index.html) and click go 2.Click on My Account dropdown button 3.Enter InValid username/email in Email text box 4.Enter Invalid password in password text box	Username: kaviyaram@gmail.com password: Testing1236786867868768 76

	5.Click on login button	

Expected Result	Actual Result	Status	Commnets
Login/Signup popup should display	Working as expected	Pass	
Application should show below UI elements: a.email text box b.password text box c.Login button with orange colour d.New customer? Create account link e.Last password? Recovery password link	Working as expected	Pass	
User should navigate to user account homepage	Working as expected	Pass	

Application should show 'Incorrect email or password ' validation message.	Working as expected		
Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	

Application should show 'Incorrect email or password ' validation message.	Working as expected	Pass	

TC for Automation(Y/N)	BUG ID	Executed By
Y		Kaviya.R, Saranya.G

Y		Saranya.G, Kaviya Varshini.S
Y		Cathlene Hena.J, Kaviya Varshini.S
Y		Cathlene Hena.J, Kaviya.R
Y		Saranya.G, Kaviya Varshini.S

Y		Cathlene Hena.J, Saranya.G

Acceptance Testing
UAT Execution & Report Submission

Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Skills/Job Recommender Application project at the time of the release to User Acceptance Testing (UAT).

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8

Totals	24	14	13	26	77

Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS

Response Time (ms)

	Total	OK	KO
Min			
50th percentile			
75th percentile			
95th percentile			
99th percentile			
Max			
Mean			
Standard Deviation			

Response Time (ms)

	Total	OK	KO
Min			
50th percentile			
75th percentile			
95th percentile			
99th percentile			
Max			
Mean			
Standard Deviation			

ExecutionsResponse Time (ms)TotalOKKO% KOCnt/sMin50th pct75th pct95th pct99th pctMaxMeanStd Dev

Gatling Version Version: 3.8.4 Released: 2022-09-13
Run Information
Date: 2022-11-17 08:58:09 GMT Duration: 1m 11s Description: fhg
StatsFixed heightFull size
Expand all groups Collapse all groups

Requests	Executions					Response Time (ms)						
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev

10. ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

- Probabilistic hybrid approach
- Proactive job recommender system.
- Semantic matchmaking for job recruitment
- Fuzzy multiple criteria method for recruitment.
- Adaptive system.
- Use many attributes.
- Use ontology to categorize jobs and as a knowledge base to define features

DISADVANTAGES:

- Key words search method.
- One way recommendation.
- Knowledge acquisition and knowledge engineering problems.
- No relational aspects are included.
- Binary representation only.
- Less attributes used.
- No perfect measures
- No relational aspects are included.
- Scalability, ramp-up, and data sparsity problems.

11. CONCLUSION

This framework facilitates the understanding of job recommendation process as well as it allows the use of a variety of text processing and recommendation. Its indispensable to have an application that matches our skills to job. Our application does that using job recommender API. Jobseekers are benefitted by knowing various job fields that would suite their skillset. On the other side, Organizations can recruit people who are knowledgeable and skilful. Users can upload their CV and input skill-set to find job postings requiring similar skill-set. To facilitate user input, the skill-set input bar is categorized into different buckets, including education, major, programming skills, business intelligence and big data skills.

12. FUTURE SCOPE

Incorporating ‘time’ into a recommender system is important, because there are often preference seasonal effects. For example, it is likely that in December, more people are going to be watching holiday-themed movies and buying home decorations. To see what would happen if a customer was shown a sub-optimal recommendation. This is taking a reinforcement learning approach, since the goal in this case would be to show jobseekers a recommendation, and then record what the jobseeker does. At times, jobseekers can be recommended something that does not seem like the best option, just to see how the jobseeker reacts which will improve the learning in the long-term. Some of the applications include being able to anticipate seasonal purchases based on recommendations, determine important notifications, and give better recommendations to users, which can increase retention and brand loyalty.

13.APPENDIX

Source Code(Git Hub): https://github.com/IBM-EPBL/IBM-Project-12037-1659367523.git

Demo:
https://drive.google.com/file/d/1kupq95Uht4jtj9WICAr2Wo5vIeyjVY76/view?usp=sharing