PROJECT DOCUMENTATION

TEAM ID	**PNT2022TMID25938**
TEAM NAME	**DareDevils**
PROJECT NAME	**Skill/Job Recommender Application**

Team Members:

1.Vidhya Sri.M

2.Swetha.S

3.Tejeswini.S

4. Kavya.K

INDEX

1. INTRODUCTION:

1.1 Project Overview:

One of the major problems that individuals face after their graduation is job search. Moreover, finding the job that better suits our skill sets is crucial as it deals with our career growth. 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. All the user needs to do is to provide the required relevant and valid details for the queries asked by the chatbot in order to proceed further with job search process. This project is aimed 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. Through this application, users can get benefitted by interacting with the chatbot and get the recommendations based on their skills. A job search API is used to get the current job openings in the market which will fetch the data directly from the webpage.

1.2 Purpose:

The importance of a career in life is huge so choosing a career needs to be done very carefully, as one's future mainly depends on the decision one takes. Also taking up a career is very common in everybody's life, but picking the right career is what makes one's life successful. Though, there are many important factors in choosing a career, the most vital one would be choosing a career that matches with our acquired skill sets in order to become a valuable asset to the organization we work in.This project is aimed to develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The project focuses mainly on providing the user with the jobs on the basis of skill set information provided by the users. To accomplish this, the use of chatbot is integrated with the application to better serve the user effectively.

2. LITERATURE SURVEY

2.1 Existing System:

The job recommending systems of today propose a dynamic user profile-based recommendations for the job seekers to address the challenge that the job applicants are facing now-a-days. Since, recently job recommendation has attracted a lot of research attention, the aim being to get a sorted list of relevant candidates for an applicant (job seeker or recruiter) using a mobile application or website is highly used. The task of job recommendation has been invariably solved using either a filter based technique or through recommender systems where categorical features associated with jobs and candidates are used to generate recommendations. In particular, the statistical results of basic features in the applied jobs are used to update the job applicants with an addition of feature selection which is employed in the text information of jobs that are applied by the job applicants for extending the feature. The application update and extend the user profile dynamically based on the historical applied jobs and behaviours of job applicants. Numerous studies have determined that there is a direct correlation between the success achieved in a particular job role and the personality traits that an individual possesses. The identification of skills relevant to a job role is done through the application of data analytics to job listings on web portals. The model has been developed and deployed in a real-world job recommender system and the best performance of the click-through rate metric has been achieved through a blend of machine learning and non-machine learning recommendations. This study provides us the evidence that identifying suitable job roles for an individual who seeks a career in the IT sector by analysing their personality.

2.1.1 Limitations:

The following are the limitations of the existing systems

i. The context formed in the peak season and the off season has an influence on the job desire of a job applicant.

ii. The choices, or even interpretations, must be made by the designer, which is not necessarily obvious, and therefore requires the use of an expert

iii. The application which uses self-reports to gain information to provide reliable results can face one of the main issues with self-reports are that the participants can lie.

iv. And not all of the applications that are in present has a chatbot or assistance to guide the user while using the application.

2.2 References

i. Hong, W., Zheng, S., & Wang, H. (2013, April). Dynamic user profile-based job recommender system. In 2013 8th International Conference on Computer Science & Education (pp. 1499-1503). IEEE.

ii. Guedj, M. (2016, August). Levelized taxonomy approach for the job seeking/recruitment problem. In 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES) (pp. 448-451). IEEE.

iii. Mirza, I. A., Mulla, S., Parekh, R., Sawant, S., & Singh, K. M. (2015, February). Generating personalized job role recommendations for the IT sector through predictive analytics and personality traits. In 2015 International Conference on Technologies for Sustainable Development (ICTSD) (pp. 1-4). IEEE.

iv. Nigam, A., Roy, A., Singh, H., & Waila, H. (2019, December). Job recommendation through progression of job selection. In 2019 IEEE 6th

International Conference on Cloud Computing and Intelligence Systems (CCIS) (pp. 212-216). IEEE.

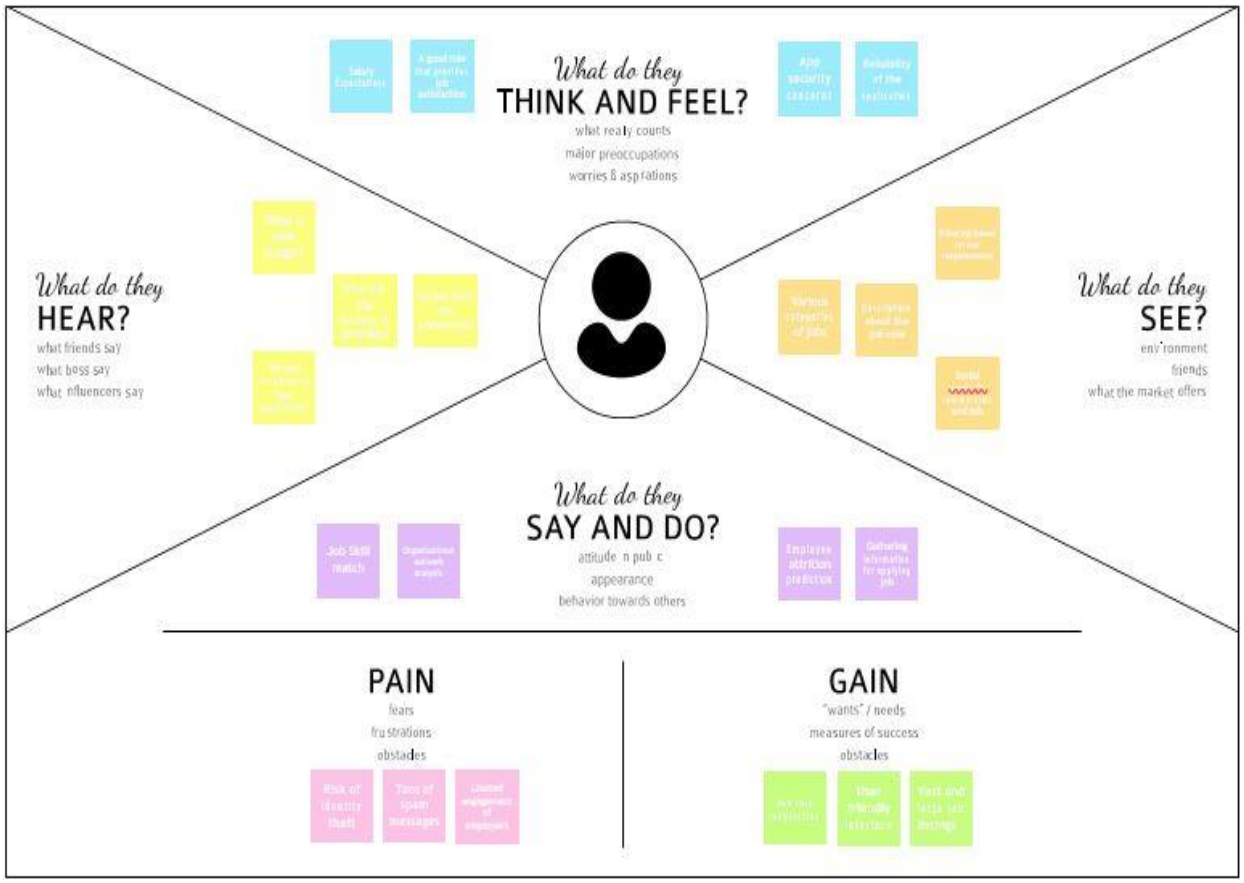
2.3 Problem Statement:

The customer is a person who is trying to find a job that can satisfy his/her job expectations like job satisfaction, salary, job role etc., to get placed in an organization where he/she can establish their skills and knowledge. But they mainly worry about losing important information leading to identity theft since, there are loads of information revealed to the public for the employers to see. This gives the user a fear about personal details theft. So, here the customer is a fresher or a working professional seeking for a new job for achieving their primary goal of getting placed in a suitable job role that can be provided by the application's job recommendation platform. Eventually, the user or customer might find difficulties in filling out their profile details or might fail to provide the required information completely. This can be a problem because to apply for a job, the profile needs to be appealing to the employers and prospective recruiters to recruit them. The resume is the vital tool in a job search since it provides you with a page or two to highlight one's relevant abilities and attributes. This is yet another problem as there is a chance for the employer to ignore one's resume if it lacks the proper format needed or improperly structured. This makes the user feel dejected on rejection of the job application. These are the problems associated with the job recommender application.

3. IDEATION AND PROPOSED SOLUTION :

3.1 Empathy Map Canvas :

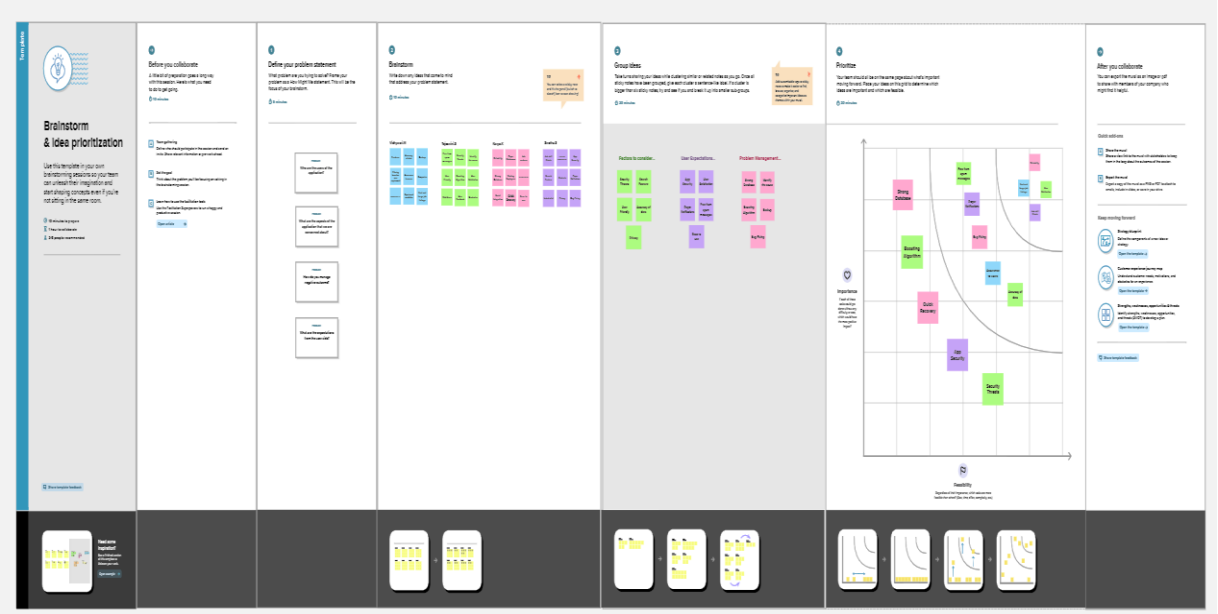
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



3.2 Ideation And Brainstorming :

Ideation is the process where you generate ideas and solutions through sessions such as Sketching, Prototyping, Brainstorming, Brainwriting, Worst Possible Idea, and a wealth of other ideation techniques. Ideation is also the third stage in the Design Thinking process. Ideation is often the most exciting stage in a Design Thinking project, because during Ideation, the aim is to generate a large quantity of ideas that the team can then filter and cut down into the best, most practical or most innovative ones in order to inspire new and better design solutions and products.

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

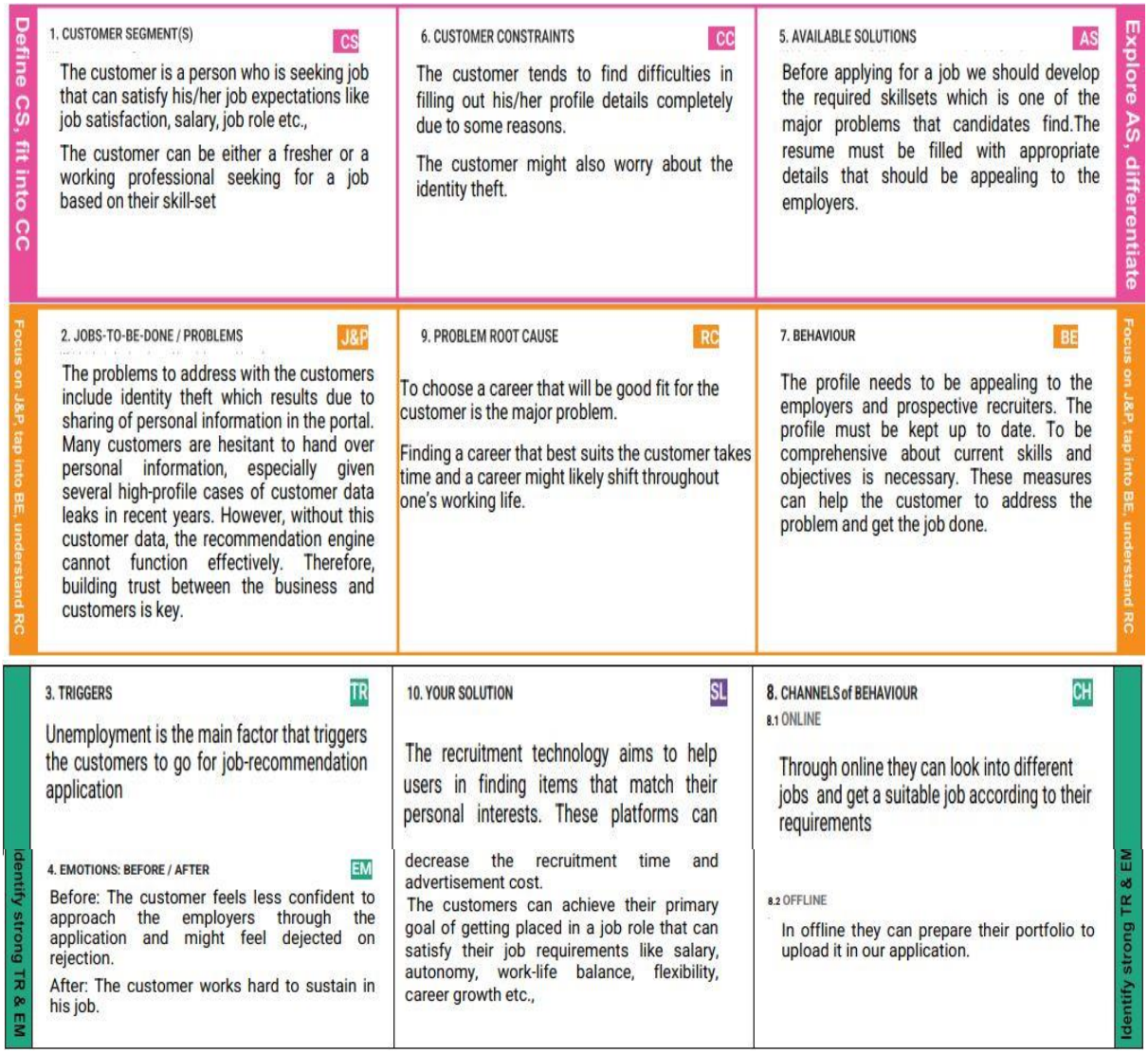


3.3 Proposed Solution :

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	The Internet-based recruiting platforms are becoming a primary recruitment channel in most companies. These platforms can decrease the recruitment time and advertisement cost. The recruitment technology aims to help users in finding items that match their personal interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. The skill recommender solution can help the fresher or the skilled person to log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.
2.	Idea / Solution description	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.	Novelty / Uniqueness	The chatbot facility will enable the user to get job recommendations based on their skill set. The job search API is used to get the current job openings in the market which will fetch the data directly from the webpage.

4.	Social Impact / Customer Satisfaction	The customers are freshers or working professionals who are trying to get placed in an organization where they can establish their skills and knowledge. They can achieve their primary goal of getting placed in a suitable job role which can be provided by the app's job recommendation platform.
5.	Business Model (Revenue Model)	Social media is one of the ways by which we can advertise about our application. Clustering and targeting the freshers through college and universities.
6.	Scalability of the Solution	Scalable recruiting is that ability to fluctuate with hiring demands is a complicated issue. It involves many building block • Establish targeted goals • Build an attractive employer brand • Build a pipeline of talent • Generate accuratjob descriptions

3.4 Problem Solution Fit :



4. REQUIREMENT ANALYSIS:

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirements (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login via Email Login via OTP
FR-4	Employer Login	Login via Email Login via OTP
FR-5	User Profile Details	Fill the profile details • Resume • Key skills • Education • Employment • Project Summary • Accomplishments
FR-6	Employer Profile Details	Fill the profile details • Resume • Career profile • Experience • Job History
FR-8	User Job Application	Upload profile details and resume Check for available jobs Skill set and Job role matching Getting recommendation for jobs Apply for job

4.2 Non-functional Requirements:

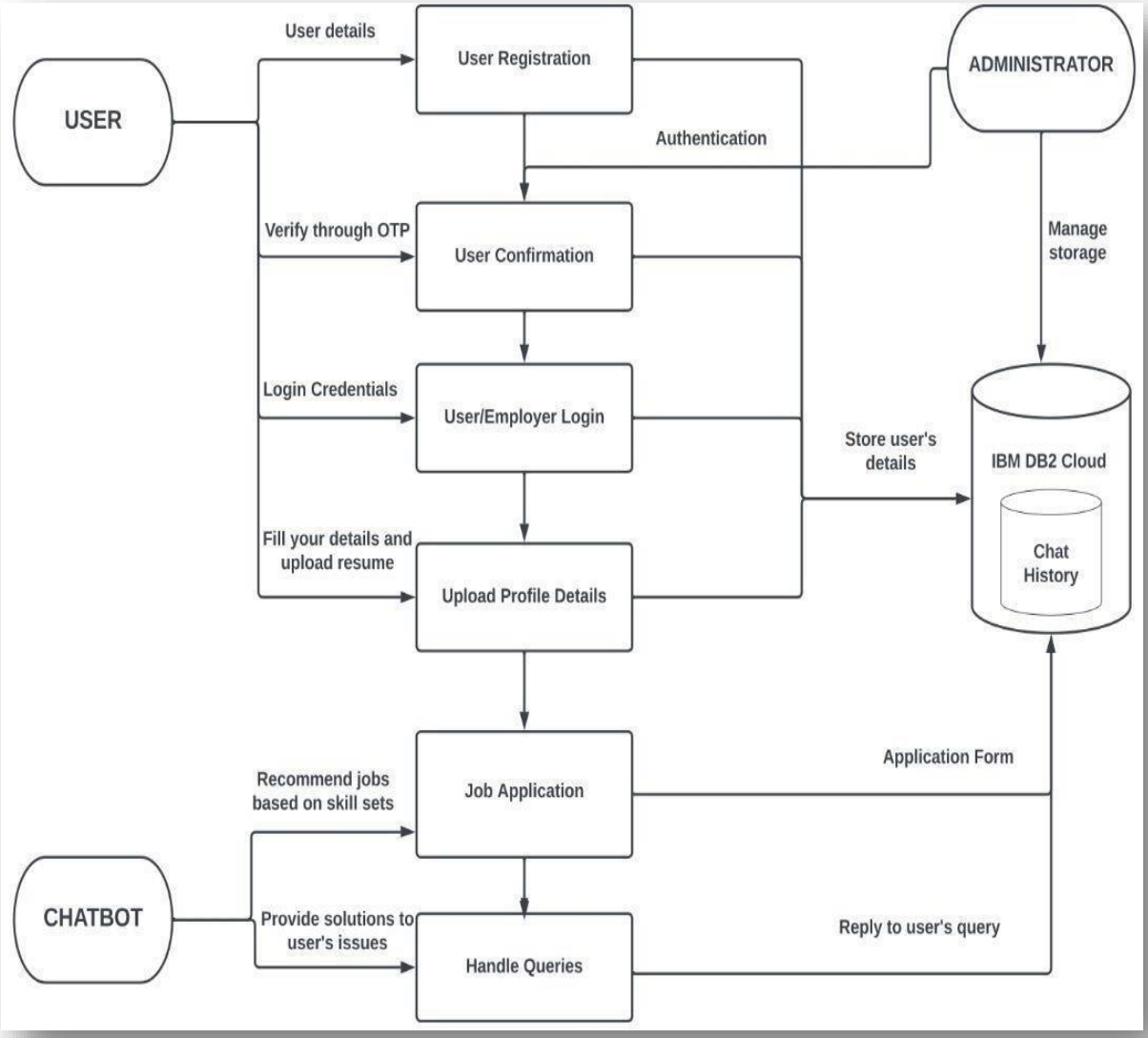
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirements	Description
NFR-1	**Usability**	It involves watching a group of users interacting with your website or application in order to see what works and what doesn't. Doing so will allow you to make small tweaks that often go overlooked in having any impact at all.
NFR-2	**Security**	Providing a secure platform to users is crucial as it involves employers and employees upload their profile details on the application
NFR-3	**Reliability**	The application exhibits the quality of being trustworthy or of performing consistently well.
NFR-4	**Performance**	Networking ensures better performance as it an effective of finding a new job. It enables for the employees to connect with the employers working in the organization of their interest to get more information of the job and the roles available.
NFR-5	**Availability**	To make sure that the application is easily available and ready to use for the users.
NFR-6	**Scalability**	The measure of the application's ability to increase or decrease in performance and cost in response to changes in the application and system processing demands.

5.PROJECT DESIGN

5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leavesthe system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

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 openingsin the market which will fetch the data directly from the webpage.

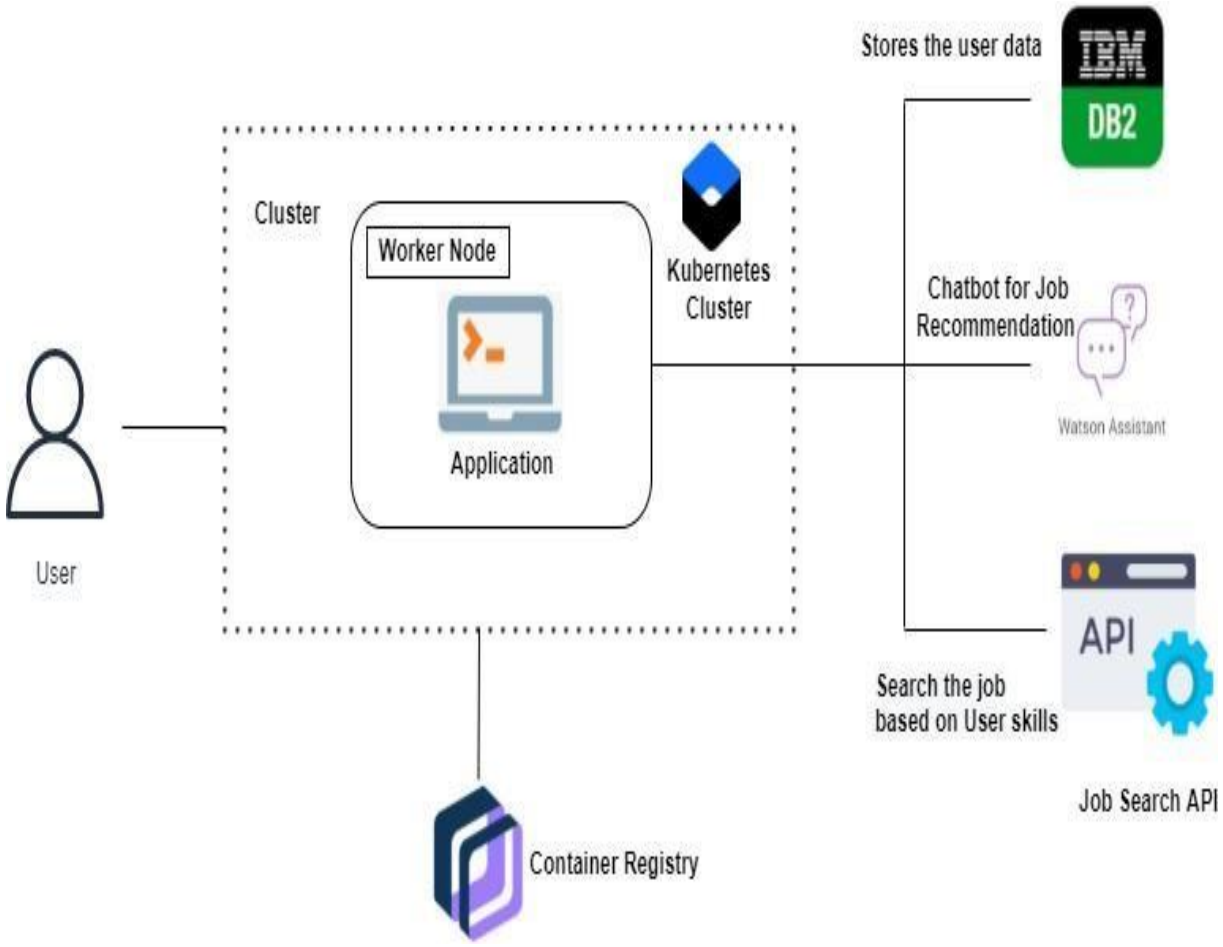


Table-1: Components & Technologies:

S.no	Component	Description	Technology
1.	User Interface	How user interacts with application e.g.,Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript,Flask, Python
2.	Application Logic-1	Logic for a process in the application	Python or Java
3.	Chatbot	To provide job recommendation and to solve user queries related to job	IBM Watson Assistance
4.	Cloud Database	To store user data and job related data	IBM DB2.
5.	File Storage	To store user data like resumes and job posts	IBM Cloud Object Storage
6.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
7.	External API-2	Purpose of External API used in the application	Job Search API
8.	Cloud Server	To Deploy the application	Kubernetes

Table-2: Application Characteristics:

S.no	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	HTML, CSS, JavaScript, Flask, Kubernetes, Docker
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	IBM DB2 IBM Cloud Object Storage
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Microservices)	Kubernetes IBM DB2
4.	Availability	Justify the availability of application	Kubernetes

5.3 User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password and confirming password.	I can access my account / dashboard	High	Sprint-1
	Confirmation	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 Google account	I can register & access the dashboard with Google account Login	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login into the application	High	Sprint-1
	User	USN-5	As a user, I can	I can access	High	Sprint-

	Profile Details		enter my profile details inthe application.	my profile details.		1
	Dashboard	USN-6	As a user , I can see my personal details and activities.	I can check and updatemy profile details	High	Sprint-1
Customer (Web user)	User Authenticati on	USN-7	To enable user authenticated access and employee restricted access	App with access tocustomer private information.	High	Sprint-2
	User Validation	USN-8	To provide access to valid users	App with access to customer private information.	High	Sprint-2
Customer Care Executiv e	Communicati on	USN-9	To communicate the how's and why's regarding service expectations within a company.	To assist customer issues and handle their queries	Mediu m	Sprint-2
Administrat or	Device Managemen t	USN-10	To create and manage users and enforce device security policies.	To manage devices	Mediu m	Sprint-2

6.PROJECT PLANNING AND SCHEDULING

6.1. Sprint planning and Estimation:

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application byentering my email, password, and confirming my password.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email onceI have registered for the application.	3	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-1		USN-3	As a user, I can register for the application through Google account	2	Low	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-1	Login	USN-4	As a user, I can login to the application byentering email and password	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-1	User Profile Details	USN-5	As a user, I can enter my profile details in theapplication.	3	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-1	Dashboard	UNS-6	As a user, I can see my personal detailsand activities.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-2	User Profile	USN-7	As a user, I can update and edit my details	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-2	Database	USN-8	As an administrator, I can store the details ofthe user in IBM database.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-2	Cloud Storage	USN-9	As a user, I can upload my resume , photo and other details.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-2	ChatBot	USN-10	As a user, I can ask the Chatbot about the job openings, which will help get recent job openings based on my skill sets.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-2	Job Search	USN-11	As a user, I can view the job posted page and apply for the suitable job.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	SendGrid Service	USN-12	As a user, I can get a notification about a job opening with the help of the SendGrid service.	5	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-3	Docker	USN-13	As a user, I can access the application.	5	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-3	Kubernetes	USN-14	As a user, I can access the application.	5	Medium	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-3	Customer care service	USN-15	As a user, I can get technical support through customer care to handle any queries.	5	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K

Sprint	Functional Requirement	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-4	Unit Testing	USN-16	As a user, I can access and utilize the application without any interruption.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-4	Integration Testing	USN-17	As a user, I can access and utilize the application without any interruption.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-4	System Testing	USN-18	As a user I can access and utilize the application without any interruption.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-4	Acceptance Testing	USN-19	As a user, I can access and utilize the application without any interruption.	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K
Sprint-4	Deployment	USN-20	Deploying the application	4	High	VidhyaSri.M,Swetha.S Tejeswini.S, Kavya.K

6.2 Sprint Delivery Schedule:

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

Velocity:

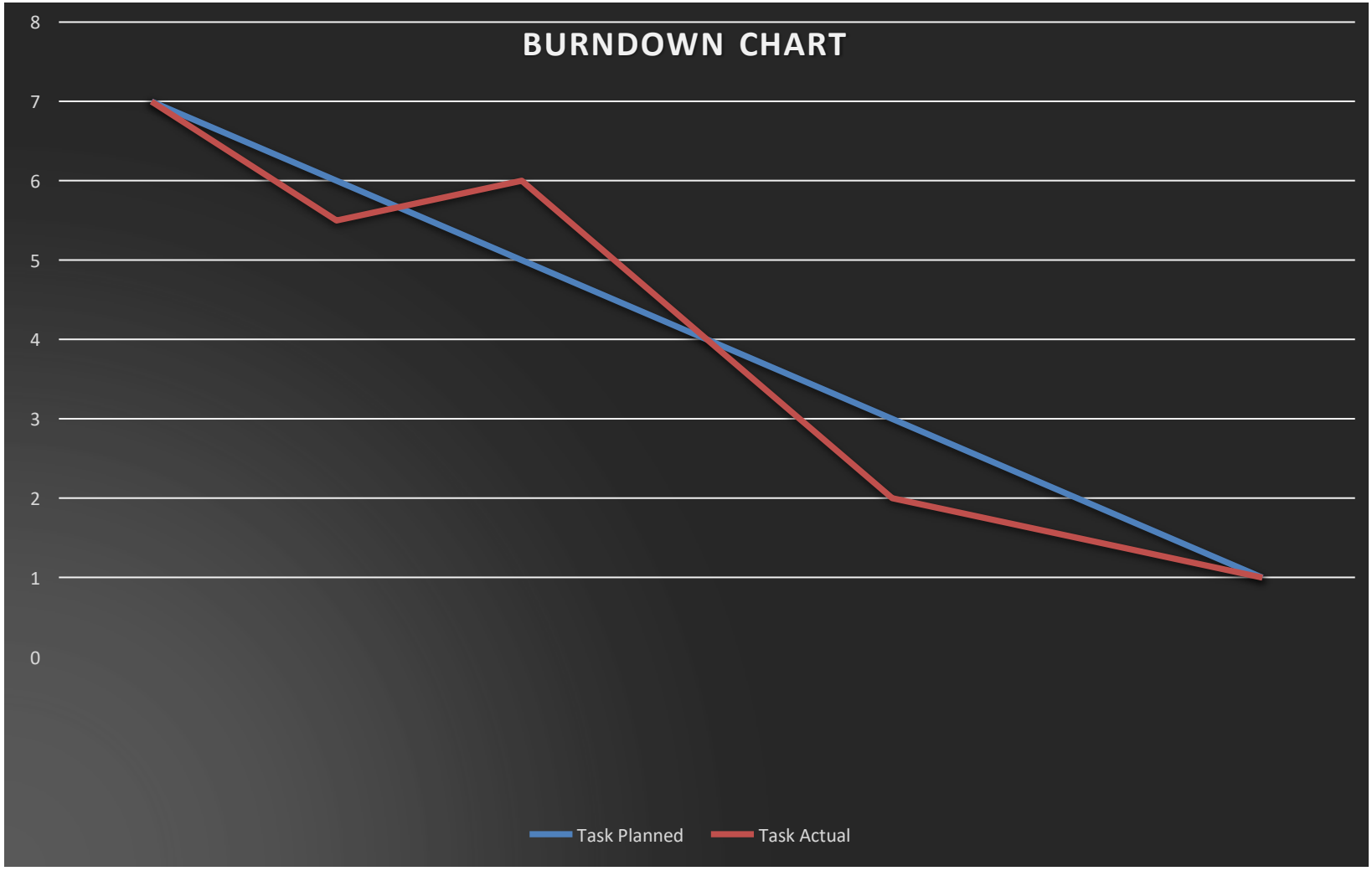
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) periteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

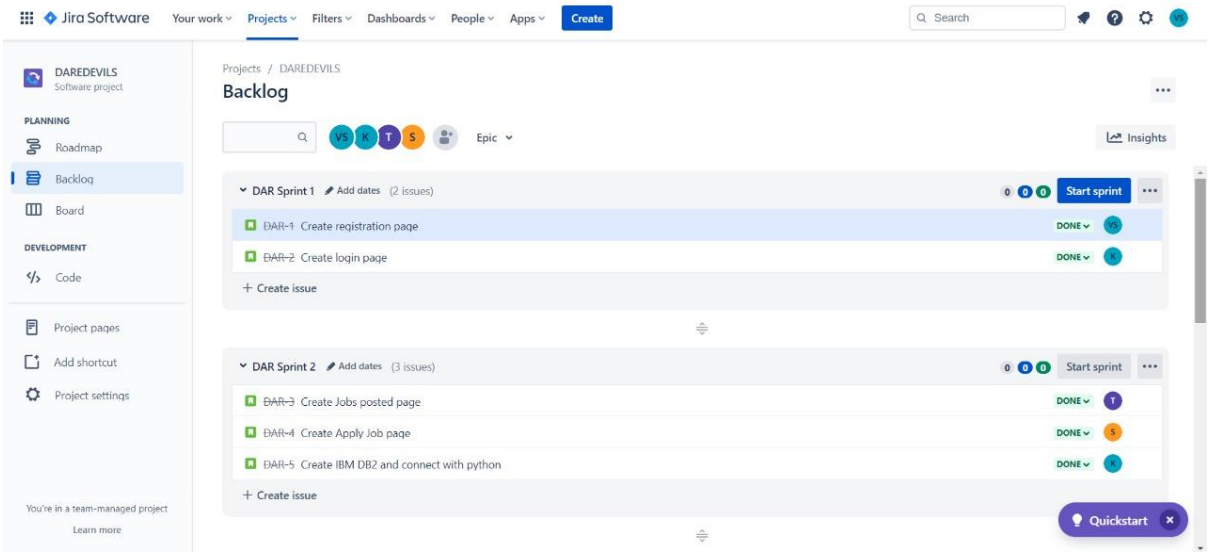
Burndown Chart:

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

	Initial Estimate	24-Oct	25-Oct	26-Oct	27-Oct	28-Oct	29-Oct
Sprint number	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6
Sprint-1	20	0	10	5	3	1	1
Sprint-2	20	2	10	4	1	1	2
Sprint-3	20	5	5	5	5	0	0
Sprint-4	20	3	3	3	3	3	5
Task planned	7	6	5	4	3	2	1
Task Actual	7	5.5	6	4	2	1.5	1


BURNDOWN CHART
8
7
6
5
4
3
2
1
0
Task Planned
Task Actual

6.3 Reports from JIRA:



7.CODING AND SOLUTIONING

7.1 Feature 1:

When you open the healthcare chatbot, the authenticate page appears first where you can authenticate with your credentials. If you are an incipient utilizer then the system prompts to engender an incipient profile asking for designation, mobile number, and password and email id. Your profile is engendered and the information is safe. It then redirects you to the authenticate page. The web application is designed in an alluring manner where the robot is moving and the colors of the background are effulgent which amends one's mood. This will magnetize a sizably voluminous number of users. This is achieved by utilizing HTML for the format and CSS for styling and FLASK was utilized for connecting the code to the web application and present it to the utilizer. Once you have authenticated in, the chatbot greets you and interaction is commenced. It will ask you a bunch a question and then depending on your symptom it explicates the disease briefly, prescribes the medicine and whether to consult a medico or not. We had accentuated on the conception that we wanted to avail unlettered people so if the utilizer is inculcated to only some extent the chatbot has text to verbalization function which reads the conversation to the utilizer for better

understanding. We have utilized the chatterbot package; it utilizes the verdant Bayesian algorithm to determine if the input verbal expression meets a particular set of criteria that warrant a replication to be engendered from that logic adapter. The chatbot functions in multiple languages for people from different regions to utilize it. Google API is utilized (gTTS) for this.

7.2 FEATURE 2:

To utilize the chatbot the utilizer must have a computer or phone to access it. It is asking to other chatbots in authenticate in and the for incipient users, they have to engender an incipient account so their details are stored safely with us for future references and it will be more facile for them to authenticate again if compulsory. After authenticate in the chatbot will ask a series of questions following which it can provide the felicitous diagnosis. For the chatbot we have not used traditional datasets, we have inscribed .yml files and have them indicted for the chatbot manually.

FLOW CHART WORKING

i. Start the chatbot application.
ii. Enter the authenticate in credentials.
iii. If new user create account. Then Step2
iv. After authenticate in conversation commences.
v. Chat with the chatbot.
vi. Few questions will be asked by the chatbot.
vii. The chatbot processes the input from the utilizer.
viii. Comparison of symptoms mentioned by the utilizer and the database.
ix. Provide precise and copacetic prescription and report.
x. We now ken what to do about the symptoms and then close the application.

8. TESTING:

8.1 Test Cases:

UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

- **Valid Input** : identified classes of valid input must be accepted.

- **Invalid Input** : identified classes of invalid input must be rejected.
- **Functions** : identified functions must be exercised.
- **Output** : identified classes of application outputs must be exercised.
- **Systems/Procedures** : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a

definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

8.2 User Acceptance Testing:

Final Stage, before handling over to the customer which is usually carried out by the customer where the test cases are executed with actual data. The system under consideration is tested for user acceptance and constantly keeping touch with the prospective system user at the time of developing and making changes whenever required. It involves planning and execution of various types of tests in order to demonstrate that the implemented software system satisfies the requirements stated in the requirement document.

This final stage is important in getting software ready for deployment and delivery.

Two set of acceptance test to be run:

i. Those developed by quality assurance group.

ii. Those developed by customer.

9. RESULTS :

9.1 Performance Metrics :

The following are the key points for a performance metrics

i. User Satisfaction / Apdex Scores

The application performance index, or Apdex score, has become an industry standard for tracking the relative performance of an application. It works by specifying a goal for how long a specific web request or transaction should take.

ii. Average Response Time

The fore mentioned user satisfaction Apdex scores as a preferred way to track overall performance. That said, averages are still a useful application performance metric.

iii. Error Rates

The last thing you want your users to see are errors. Monitoring error rates is a critical application performance metric. It is common to see thousands of exceptions being thrown and ignored within an application. Hidden application exceptions can cause a lot of performance problems.

iv. Count of Application Instances

If your application scales up and down in the cloud, it is important to know how many server/application instances you have running. Auto-scaling can help ensure your application scales to meet demand and saves you money during off-peak times. This also creates some unique monitoring challenges.

v. Request Rate

Request rates can be useful to correlate to other application performance metrics to understand the dynamics of how your application scales. Monitoring the request rate can also be good to watch for spikes or even inactivity. If you have a busy API that suddenly gets no traffic at all, that could be a really bad thing to watch out for.

vi. Application & Server CPU

If the CPU usage on your server is extremely high, you can guarantee you will have application performance problems. Monitoring the CPU usage of your server and applications is a basic and critical metric. Virtually all server and application monitoring tools can track your CPU usage and provide monitoring alerts. It is important to track them per server but also as an aggregate across all the individually deployed instances of your application.

vii. Application Availability

Monitoring and measuring if your application is online and available is a key metric you should be tracking.

viii. Garbage Collection

If your application is written in .NET, C#, or other programming languages that use garbage collection, you are probably aware of the performance problems that can arise from it. When garbage collection occurs, it can cause your process to suspend and can use a lot of CPU.

10. ADVANTAGES AND DISADVANTAGES :

10.1 Advantages :

i. The recommender application 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.

ii. All the job boards would provide us with the jobs that the user searched for based on keywords that we entered in that search box but in our application we provide recommendation for jobs based on user skills and preferences.

iii. New recommendation of jobs can also be made when there is a change in user preference, i.e. if a user thinks to change his/her job domain by updating his new skills that can also be implemented in our application.

iv. The application we are creating also provide the user with chatting assistant which enables them to get a more accurate jobs based on their skillset.

10.2 Disadvantages :

i. There are loads of personal information that you have to display on your profile for prospective employers to see. Hence, in a case whereby servers develop an issue, you stand a risk of losing important information to the public, resulting in identity theft.

ii. In social network websites like linkedin they require you to put up an attractive profile. That is a profile that is appealing to employers and prospective recruiters. People however find it hard to fill out profile details completely due to one reason or the other.

iii. It can be difficult to measure their effectiveness since not all online recruitment services offer an in-depth analysis of the user's posting.

iv. Some applications might be fake to get information about the company or hiring officer. Eventually, spammers might use that info to misuse.

11. CONCLUSION:

In conclusion, the three main objectives of our apps is User Growth, Networking, Job recommendation. The application we are creating also provide the user with chatting assistant which enables them to get a more accurate jobs based on their skillset. The recommender application 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. Atlast every user will be recommended by a suitable job. Also taking up a job is very common in everybody's life, but picking the right job is what makes one's life successful. Recommending a right job to user is achieved by our application.

12.Future Scope:

Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation. 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. In addition, we plan to enhance the similarity measures that suitable for this problem.

13. APPENDIX

13.1 Source Code

Registration page

```
<html>
<head>
   <title>Registration</title>
   <link rel="stylesheet"
    href="{{url_for('static',filename='css/bootstrap.min.css')}}">
   <link rel="stylesheet"
    href="{{url_for('static',filename='myfont/css/all.min.css')}}">
   <script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
   <style>
      .shadow{
         box-shadow:3px 3px 10px black;
         padding:30px;
      }
        .button{
        background-color:#0161FF ;
        color:white;
        padding:20px 10px;
        text-align:center;
        font-size:15px;
                cursor:pointer;
                border-radius:50%;
        }
   </style>
</head>
<body style="margin:0;
    background-image:url('/static/job.png');
```

```
background-repeat:no-repeat;
background-position:center;
background-attachment:fixed;">
<div class="container" style="margin-top:50px">
   <div class="col-md-offset-3 col-md-5">
      <div class="shadow">
         <form method="POST" action="/register" class="page-align">
            <h2 class="page-header text-primary text-
center">Registration</h2>
            <div class="form-group">
               <label>Name</label>
               <input type="text" class="form-control" name="name"
required>
            </div>
             <div class="form-group">
               <label>Mail</label>
               <input type="text" class="form-control" name="mail" required>
             </div>
            <div class="form-group">
               <label>Contact</label>
               <input type="text" class="form-control" name="contact"
required>
            </div>
            <div class="form-group">
               <label>Password</label>
               <input type="password" class="form-control" name="password"
required>
            </div>
             <div class="form-group">
```

```
                    <input type="submit" class="btn btn-success btn-block"
    value="Register">
                     <a href="{{url_for('index')}}" class="btn btn-primary btn-
    block">Back to Home</a>
                </div>
            </form>
        </div>
      </div>
   </div>
</body>
</html>
<script>
   $(document).ready(function(){
      $("form").attr("autocomplete","off");
   });
</script>
```

Login page

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Index</title>
   <link rel="stylesheet"
    href="{{url_for('static',filename='css/bootstrap.min.css')}}">
   <link rel="stylesheet"
    href="{{url_for('static',filename='myfont/css/all.min.css')}}">
   <script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
```

```
    <style>
      .shadow{
         box-shadow:3px 3px 10px black;
         padding:30px;
      }
        .button{
        background-color:#0161FF ;
        color:white;
        padding:20px 10px;
        text-align:center;
        font-size:15px;
        cursor:pointer;
        border-radius:50%;
        }
   </style>
</head>
<body style="margin:0;
    background-image:url('/static/job.png');
    background-repeat:no-repeat;
    background-position:center;
    background-attachment:fixed;">
   <div class="container" style="margin-top:100px">
      <div class="col-md-offset-1 col-md-10">
         <div class="col-md-offset-3 col-md-5">
            <div class="shadow">
            <form action="/login" method="POST" autocomplete="off" >
               <h3 class="page-header text-primary">Login</h3>
```

```
            {% with messages = get_flashed_messages(with_categories=true)
%}
        {% if messages %}
            {% for category,message in messages %}
                <div class="alert alert-{{category}}">{{message}}</div>
            {% endfor %}
        {% endif %}
        {% endwith %}

            <div class="form-group">
                <label>Username</label>
                <input type="text" name="name" class="form-control"
required>
            </div>
            <div class="form-group">
                <label>Password</label>
                <input type="password" name="password" class="form-control"
required>
            </div>
            <div class="form-group">
                <input type="submit" value="Login" class="btn btn-success btn-
block">
                <a href="{{url_for('register')}}" class="btn btn-primary btn-
block">Register</a>
            </div>
        </form>
            </div>
      </div>
  </div>
```

```
    </div>
</body>
</html>
<script>
    $(document).ready(function(){
        $(".alert").hide(3000)
    });
</script>
```

View Job Posted page:

```
    <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Users Page</title>
     <link rel="stylesheet"
href="{{url_for('static',filename='css/bootstrap.min.css')}}">
    <link rel="stylesheet"
href="{{url_for('static',filename='myfont/css/all.min.css')}}">
    <script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<style>
.button{
                background-color:#0161FF;
                color:white;
                padding:20px 10px;
                text-align:center;
                font-size:15px;
                cursor:pointer;
                border-radius:50%;
```

```
                }
.button1{
background-color:#0161FF;
color:white;
padding:20px 10px;
text-align:center;
font-size:15px;
cursor:pointer;
border-radius:50%;
}
.apply{
color:blue;
</style>
</head>
<body style="margin:0;
        background-image:url('/static/velocix.png');
        background-repeat:no-repeat;
        background-position:center;
        background-attachment:fixed;">
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:220px;
top:180px; font-size:15px;        ">Apply Here</a>
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:220px;
top:380px; font-size:15px;        ">Apply Here</a>
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:220px;
top:560px; font-size:15px;        ">Apply Here</a>
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:1100px;
top:160px; font-size:15px;        ">Apply Here</a>
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:1100px;
top:410px; font-size:15px;        ">Apply Here</a>
```

```
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:1100px;
top:580px; font-size:15px;        ">Apply Here</a>
<a href="{{url_for('apply')}}" class='apply' style="position:fixed; left:600px;
top:570px; font-size:15px;        ">Apply Here</a>
<div class="container">
        <div class="col-md-offset-3 col-md-5">
                <div class="job-portal">
           <h2 class="page-header text-primary text-center">Job Seekers</h2>
           <h3 class="text-success">Welcome : {{session["name"]}}</h3>

           <hr>
          <a href="{{url_for('logout')}}" style="position:relative; left:30px;"
><button class="button">Logout</button></a>
         </div>
        </div>
     </div>

</body>
</html>
```

Apply job pages:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Job Apply Page</title>
     <link rel="stylesheet"
href="{{url_for('static',filename='css/bootstrap.min.css')}}">
```

```
    <link rel="stylesheet"
href="{{url_for('static',filename='myfont/css/all.min.css')}}">
    <script src="{{url_for('static',filename='js/jquery.min.js')}}"></script>
<style>
.button{
                background-color:#0161FF ;
                color:white;
                padding:20px 10px;
                text-align:center;
                font-size:15px;
                cursor:pointer;
                border-radius:50%;
                }
</style>
</head>
<body style="margin:0;
        background-image:url('/static/job.png');
        background-repeat:no-repeat;
        background-position:center;
        background-attachment:fixed;">
    <div class="container">
      <div class="col-md-offset-3 col-md-5">
                <div class="job-portal">
          <h2 class="page-header text-primary text-center">Candidate
Details</h2>
          <h3 class="text-success">Please Enter Your Details Below</h3>

          <hr>
```

```
                <a href="{{url_for('logout')}}" class="btn btn-primary btn-
block">Logout</a>

        <div class="shadow">
           <form method="POST" action="/register" class="page-align">
             <h2 class="page-header text-primary text-center">Details</h2>
             <div class="form-group">
                <label>Degree</label>
                <input type="text" class="form-control" name="degree"
required>
             </div>
              <div class="form-group">
                <label>Year of Passed Out</label>
                <input type="text" class="form-control" name="year of passed
out" required>
              </div>
             <div class="form-group">
                <label>Stream</label>
                <input type="text" class="form-control" name="stream"
required>
             </div>
             <div class="form-group">
                <label>Percentage or CGPA</label>
                <input type="text" class="form-control" name="percentage"
required>
             </div>
                                  <div class="form-group">
                <label>SkillSet</label>
```

```
                    <input type="text" class="form-control" name="skillset"
required>
                </div>
                                        <div class="form-group">
                    <label>Year of Experience</label>
                    <input type="text" class="form-control" name="year of
experience" required>
                </div>
                 <div class="form-group">
                    <input type="submit" class="btn btn-success btn-block"
value="Submit" onclick="myFunction()">
                </div>
              </form>
           </div>
        </div>
    </div>
          </body>
</html>
          <script>
  window.watsonAssistantChatOptions = {
    integrationID: "e49f41f2-9024-490a-adfb-4e1cac49bed6", // The ID of this
integration.
    region: "au-syd", // The region your integration is hosted in.
    serviceInstanceID: "f03e1b77-da51-4d3c-99b2-bfd44e9cdf67", // The ID of
your service instance.
    onLoad: function(instance) { instance.render(); }
  };
  setTimeout(function(){
    const t=document.createElement('script');
```

```
   t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
   document.head.appendChild(t);
  });
</script>
<script>function myFunction(){
        alert("Thanks For Applying, We Will Send You A Suitable Job Opening
Details To Your MailID");}
        </script>
```

13.2 GitHub & Project Demo Link

GitHub Link:

https://github.com/IBM-EPBL/IBM-Project-1479-1658390082

Project Demo Link:

https://screenrec.com/share/rSBaEk8zPL