SKILL / JOB RECOMMENDER APPLICATION

TEAM ID: PNT2022TMID45573

TEAM MEMBERS :

AKSHAYASRI S	(812819104003)
SNEKA J	(812819104044)
JAYASRI K	(812819104018)
SIVARANJANI S	(812819104043)

FACULTY MENTOR:

Mrs. **JAYASRI KANDHAVELAN BALAMURUGAN**

CONTENTS

S.NO TITLE PAGE NO

1. INTRODUCTION

The world has seen an important increase in the demand for Cloud-based applications. This has in turn increased the demand for Cloud application development. As a result, the past few years have had a consolidation of the Cloud computing market. Cloud apps and services are used, directly or indirectly, by almost everyone. Businesses have also increased their use of Cloud-based applications and services, even if they sometimes don’t know it. If you use SaaS tools, you are surely using a Cloud app. However, Cloud apps are more than just that. Cloud-based applications, also known as Cloud apps, seem to be taking over. In theory, a Cloud app is one that uses Cloud-based services. So, whether an app is mobile or web, they probably use some sort of Cloud service. What really differentiates a Cloud app from a native one is the extent to which they use Cloud services. Increased dependence on the Cloud’s processing power is the result of companies building innovative and creative solutions to all sorts of problems that use technology to do things that were previously impossible. Thanks to the ability to process large amounts of data (Big Data) through third party-owned IT infrastructure, companies can perform massive calculations and deliver top services. Cloud services have opened the possibility for many web-based Cloud applications, also known as web apps. A web app is one where most of the computation occurs in the Cloud, not on the device itself, and is usually built with the use of Cloud application development services. A new form of a web app, known as a Progressive Web App (PWA), is also seeing an increase in popularity. Cloud application development is the process through which a Cloudbased app is built. It involves different stages of software development, each of which prepares your app to go live and hit the market. The best Cloud app development teams use DevOps practices and tools like Kubernetes. However, an experienced app development company should ideally be technology agnostic, which means being able to build your Cloud app using any technology you prefer. Most apps built using the Cloud are highly dependent on the Cloud to operate.

1.1 PROJECT OVERVIEW

With an increasing number of cash-rich, stable, and promising technical companies/startups on the web, which are in much demand right now, many candidates want to apply and work for these companies. They tend to miss out on these postings because there is an ocean of existing systems that list millions of jobs which are generally not relevant at all to the users. There is an abundance of choices and not much streamlining. On the basis of the actual skills or interests of an individual, job seekers often find themselves unable to find the appropriate employment for themselves. This system, therefore, approaches the idea from a data point of view, emphasizing more on the quality of the data than the quantity.

1.2 PURPOSE

The purpose of Skill / Job Recommender Application is to provide information that can be used to apply for the job. In today world, Increases in population results of demand for goods and services. When there is an increase in these goods and services there is a need for workers to make or manufacture or to create the goods and services that are really a need for human habitat. So, there are lot of employment opportunities in every industry. There is unemployment only because of lack of skill set in their domain or fear of missing out on a job. When a job seeker is afraid of getting the desired job, he might losses the job which he really deserved for it. The main purpose of the job recommender application is to provide job opportunities for each and every single person. The only thing the job seeker wants to do is just to approach the application and apply for the job. He will be provided with the login credentials with the confirmation email. There he can find numerous job opportunities. He will be guided with the in-built chat bot, which guides the job seekers to apply for the job and recommends the availability of jobs based on their interest. The chat bot is built with IBM Watson Assistant that is very much helpful in collecting the job seeker interests and also guides them to apply for it.

2. LITERATURE SURVEY

Authors: Koen Bothmer and Tim Schlippe.

Title: Skill Scanner: Connecting and Supporting Employers, Job Seekers and Educational Institutions with an AI-based Recommendation System.

Published in: June 2022.

Methodology:

In this paper, all three players—employers, job seekers1 , and educational institutions—need to be aligned. There are already natural language processing (NLP) approaches to extract text data from job seekers' CVs (curriculum vitae), employers' job postings or educational institutions' learning curricula and give recommendations to one of these players. However, this way all three parties use AI in isolation from one another. For example, presents a Word2Vec-based system which informs employers how well job seekers' CVs fit job postings.

LinkedIn has a system that recommends employers' jobs to job seekers based on their personal profile. investigate how AI-based recommendations help job seekers find study programs based on their profile. use a combination of knowledge graph and BERT for helping employers find suitable candidates in a corpus of CVs. Our approach leverages similar NLP methods, but it benefits not only one, but all three players involved. Connecting and supporting them all allows the greatest possible exchange of information and satisfies their needs as illustrated in Figure 1: (1) Employers want to automatically check which of their required skills are covered by applicants' CVs (Find and Select) and know which courses their employees can take to acquire missing skills (Upskill Workforce). (2) Job seekers want to know which skills from job postings are missing in their CVs (Fit to Demand), and which study programs they can take to acquire missing skills (Find Program). (3) Educational institutions want to ensure that the skills required in job postings are covered in their curricula (Fit to Demand), recommend study programs and advise students (Advise).

LIMITATIONS:

- Requires educational institute data like syllabus, lesson plans, etc.
- Returns reports which might be tedious to read.

Authors: Jorge Martinez-Gil, Bernhard Freudenthaler, Thomas Natschläger.

Title: Recommendation of Job Offers Using Random Forests and Support Vector Machines

Published in: March 2018

Methodology:

In order to improve the accuracy of the predictions, great research efforts have been made in the last few years concerning the definition of methods for combining a number of simple methods. These methods construct a set of hypotheses (a.k.a. ensemble), and combine the predictions of the ensemble in some way to classify new data. The precision obtained by this combination of hypotheses is usually better than the precision of each individual component. One of the most popular methods in this context are random forests. On the other hand, algorithms based on n-dimensional geometry, where given a set of past solved cases from the past are also gaining popularity. In this way, it is possible to label the classes and train the algorithm to build a geometric model that correctly classifies a new sample. We give a deeper insight of these two methods below

- Random Forests
- Support Vector Machines.

LIMITATIONS:

 SVMs work with models hard to interpret by humans.
Does not use textual description from job offers.

Authors: Pradeep Kumar Roy[a], Sarabjeet Singh Chowdhary[b], Rocky Bhatia[b].

Title: A Machine Learning approach for automation of Resume Recommendation system.

Published in: IEEE (2020) Methodology:

In this paper, The aim of this work is to find the right candidates resume from the pool of resumes. To achieve this objective, we have developed a machine learning based solution, The complete framework for the proposed model is shown in Figure 2. The proposed model worked mainly in two steps: i) Prepare and ii) Deploy and Inference. Dataset Description: The data was downloaded from the online portal(s) and from Kaggle. The data is in Excel format, with three column ID, Category, and Resume. ID - The sequence number of the resume, Category - Industry sector to which the resume belongs to, and Resume - The complete CV of the candidate. The number of instances for the different domain can be seen from Figure.

LIMITATIONS

- Accuracy is only 78%.

Authors: Ravitha Mishra, Sheetal Rathi.

Title: Enhanced DSSM (deep semantic structure modelling) technique for job recommendation.

Published in: IEEE (2016)

Methodology:

In this paper, search query are used as a (query, document) pair which consist search criteria such as job title, skills, and location from the recruiter dataset the goal of the DSSM-SJR model is to select the most prominent skill and rank the candidates based on a graph-based recommendation approaches.

In the DSSM technique, the query and document text are modelled by separate neural network layer and crossed before final score computation and further it optimizes for search engagement and considers as a positive label. In feature representation, DSSM uses document text and query as member pair and converts them to character trigrams. The generated character trigram is used as an input of the proposed model. The said technique is also termed as word hashing and it represents each word in the character trigram format instead of vector representation of the entire word.

LIMITATIONS

- Words with multiple meanings are conflated into a single representation.

Authors: Naresh Kumar, Manish Gupta, Deepak Sharma.

Title: Technical Job Recommendation System Using APIs and Web Crawling.

Published in: Jun 2022 Methodology:

In this paper, This module is used to collect important deciding details about a particular company that will be used to filter the companies from the list. These parameters are important to shortlist only the best companies, so that irrelevant options are not displayed to the user. This was done by crawling and scraping the concerned web listings [27, 28] for these companies. The parameters that have been used to filter out companies are series of funding, total funding, number of investors, number of employees (organization size), unicorn status [29], and latest technology stack. Upon using these parameters, a new list of shortlisted companies has been created which has fewer results. This is the final list of this research and recommendation system [30] to work upon.

LIMITATIONS

- This paper uses collaborative filtering which faces well-known problems of privacy breaches and cold start.
- Crawling process is not automate.

2.1 Existing problem

Techniques for automatic recommendation of job offers are specifically designed to address the problem of information overload by giving priority to information delivery for individual users based on their learned preferences [1]. The most common way to process this information nowadays consists of automatically processing the documents involved in the e-recruitment process. For each document, it is possible to extract a vector for each of its fields (which contain textual information) using the bag-of-words model and TF-IDF as weighting function. Then, some kind of methods for set comparison can shed results on the suitability of a given candidate for a specific job offer. In general, most methods try to exploit solutions based on the Vector Space Model (VSM) to measure the similarity ratio between the original job offer and the application received. It is a solution easy to implement, with very low computational costs, and that tradition has achieved very good results in the context of job recommendation. However, new trends bet on the use of machine learning technology in order to overcome the traditional limitations concerning the incapability of going further beyond the syntactical representation of the document.

2.2 References

[1] Palmer, R.: Jobs and Skills Mismatch in the Informal Economy. 978-92-2-131613-8 (2017).

[2] Fernández-Reyes, F.C., Shinde, S.: CV Retrieval System Based on Job Description Matching Using Hybrid Word Embeddings, Computer Speech & Language, vol 56 (2019).

[3] Fabian Abel, Andras A. Benczur, Daniel Kohlsdorf, Martha Larson, Robert Palovics: Proceedings of the 2016 Recommender Systems Challenge, RecSys Challenge 2016, Boston, Massachusetts, USA, September 15, 2016. ACM 2016.

[4] Daniel Bernardes, Mamadou Diaby, Raphael Fournier, Francoise FogelmanSoulie, Emmanuel Viennet: A Social Formalism and Survey for Recommender Systems. SIGKDD Explorations 16(2): 20-37 (2014).

[5] Al-Otaibi S.T., Ykhlef M
A survey of job recommender systems
International Journal of Physical Sciences, 7 (2012), pp. 5127-5142.

[6] Breaugh J.A.
The use of biodata for employee selection: Past research and future directions Human Resource Management Review, 19 (2009), pp. 219-231

[7] Abualigah, Laith, Diabat, Ali, Mirjalili, Seyedali, Abd Elaziz, Mohamed, Gandomi, Amir H. 2021. The arithmetic optimization algorithm, Computer Methods in Applied Mechanics and Engineering, 376, 113609, ISSN 0045-7825.

[8] Batmaz, Z., Yurekli, A., Bilge, A., Kaleli, C. 2018. A review on deep learning for recommender systems: challenges and remedies, Artificial Intelligence Review, Aug 2018.

[9] C. Slamet, R. Andrian, D. S. A. Maylawati, W. Darmalaksana, and M. A. Ramdhani,
"Web scraping and Naïve Bayes classification for job search engine," *IOP Conference Series: Materials Science and Engineering*, vol. 288, no. 1, Article ID 012038, 2018.

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

2.3 Problem Statement Definition

Skill / Job Recommender Application provides a solution for a problem. Where the problem is unemployment and the solution for it is providing the desired job based on the job seekers' interests. Here our application provides the solution for it.

Rajesh is a fresher who needs a job and wants to know the availability in the companies for his skills because he wants to be independent.
Nisha is UI/UX designer who really wants to switch her career in other techy field so that she can gain knowledge in different domain.

Kabi is a senior analyst in specific organization needs a another a high paying job because he has a better skill in this domain with low pay.
Akash is not okay with his organization so he wants to switch organization so that he can love his work and also can learn new things in other firm.
Dharshini is a full stack developer who lost her job because of layoff so she wants to find a new job.

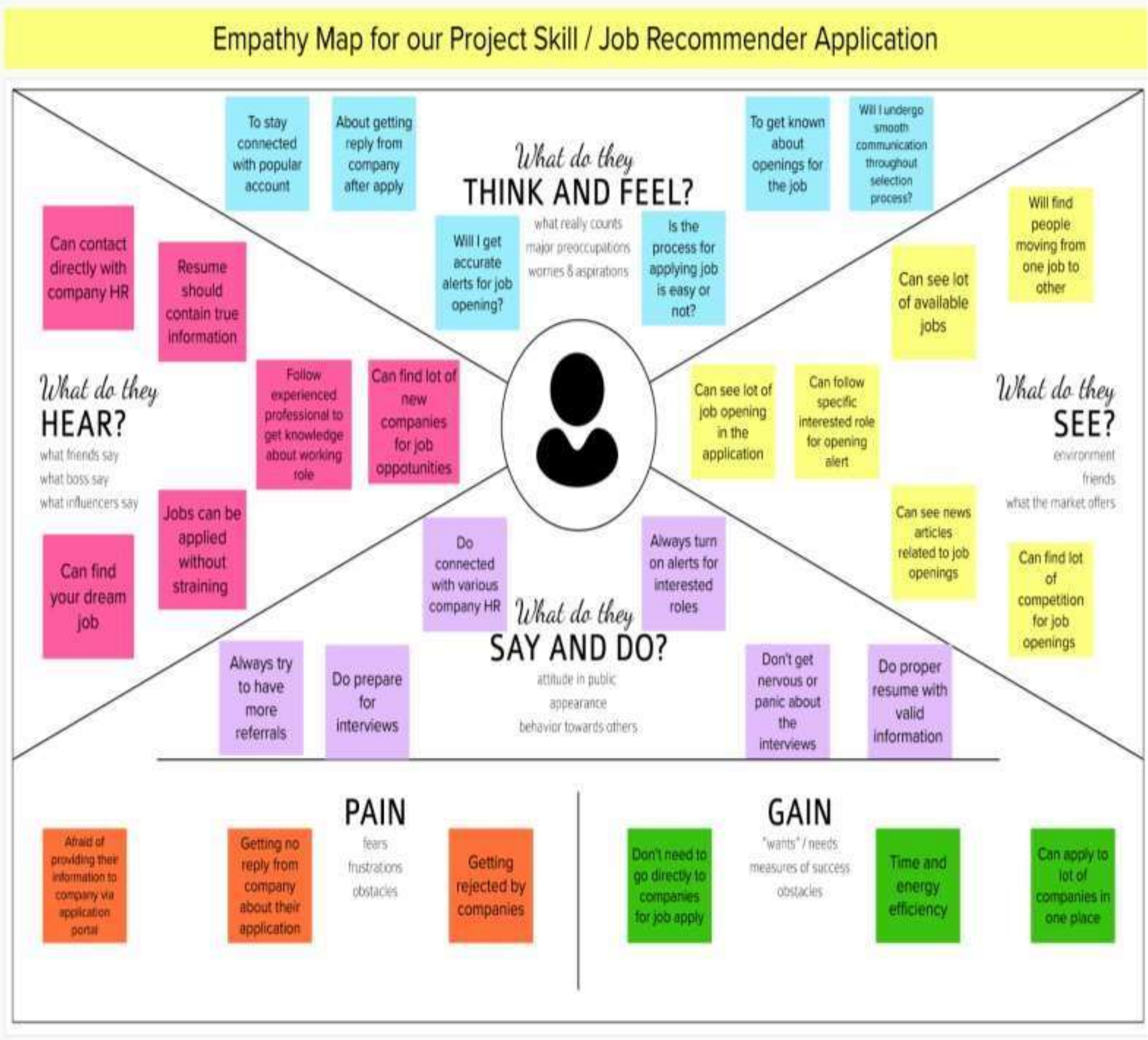
Question	Description
Who does the problem affect?	Persons who are searching for a new job.
What is the issue?	When he/she have no jobs or not okay with current job.
When does the issue occur?	While looking for job.
Where is the issue occurring?	Issue occurs when there is lot of unemployment.
Why is it important that we fix the problem?	To place the skilled employee in respected field.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community. Empathy maps can be used whenever you find a need to immerse yourself in a user's environment. Everyone would add at least one sticky to every section. You might ask questions, such as:

- What would the user be thinking and/or feeling? What are some of their worries and aspirations?
- What would their friends, colleagues, and boss be likely to say while the user is using our product? What would the user hear in these scenarios?
- What would the user see while using our product in their environment?
- What might the user be saying and/or doing while using our product? How would that change in a public or private setting?
- What are some of the user’s pain points or fears when using our product?
- What gains might the user experience when using our product?



Empathy Map for our Project Skill / Job Recommender Application

What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

To stay connected with popular account

About getting reply from company after apply

Will I get accurate alerts for job opening?

Is the process for applying job is easy or not?

To get known about openings for the job

Will I undergo smooth communication throughout selection process?

What do they HEAR?

what friends say
what boss say
what influencers say

Can contact directly with company HR

Resume should contain true information

Follow experienced professional to get knowledge about working role

Can find lot of new companies for job oppotunities

Jobs can be applied without straining

Can find your dream job

What do they SEE?

environment
friends
what the market offers

Will find people moving from one job to other

Can see lot of available jobs

Can see lot of job opening in the application

Can follow specific interested role for opening alert

Can see news articles related to job openings

Can find lot of competition for job openings

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

Do connected with various company HR

Always turn on alerts for interested roles

Always try to have more referrals

Do prepare for interviews

Don't get nervous or panic about the interviews

Do proper resume with valid information

PAIN

fears
frustrations
obstacles

Afraid of providing their information to company via application portal

Getting no reply from company about their application

Getting rejected by companies

GAIN

"wants" / needs
measures of success
obstacles

Don't need to go directly to companies for job apply

Time and energy efficiency

Can apply to lot of companies in one place

EMPATHY MAP CANVAS

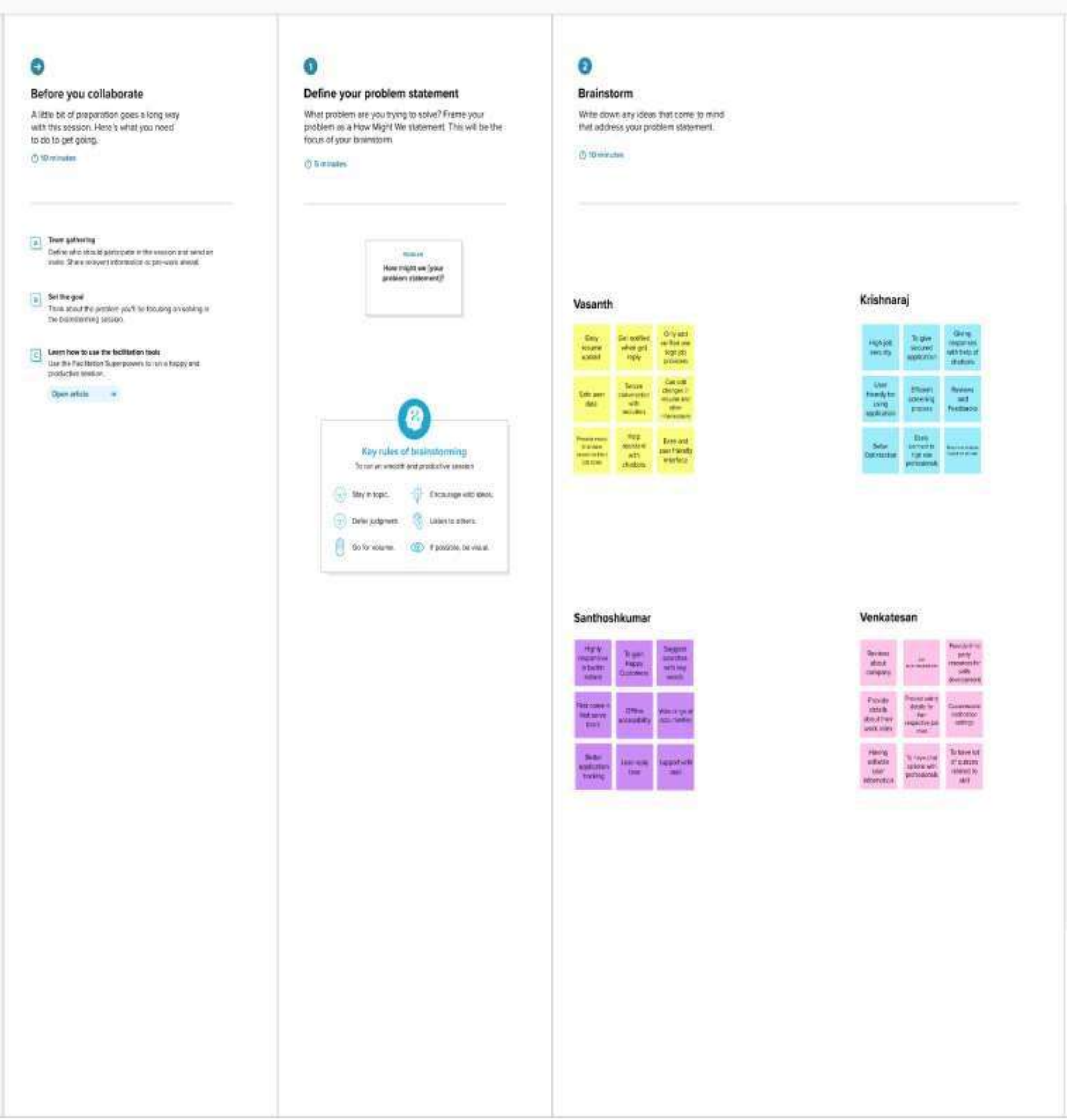
3.2 Ideation and brainstorming

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity. Brainstorming is usually conducted by getting a group of people together to come up with either general new ideas or ideas for solving a specific problem or dealing with a specific situation.

For example, a major corporation that recently learned it is the object of a major lawsuit may want to gather together top executives for a brainstorming session on how to publicly respond to the lawsuit being filed.

Participants in a brainstorming session are encouraged to freely toss out whatever ideas may occur to them. The thinking is that by generating a large number of ideas, the brainstorming group is likely to come up with a suitable solution for whatever issue they are addressing.

The lines between ideation and brainstorming have become a bit more blurred with the development of several brainstorming software programs, such as Bright idea and Idea wake. These software programs are designed to encourage employees of companies to generate new ideas for improving the companies' operations and, ultimately, bottom-line profitability.



Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM
How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Vasanth

Easy resume upload
Get notified when got reply
Only add verified user login job providers
Safe user data
Secure communication with recruiters
Can edit changes in resume and other information
[illegible]
Help desk/assistance with chatbots
Easy and user friendly interface

Krishnaraj

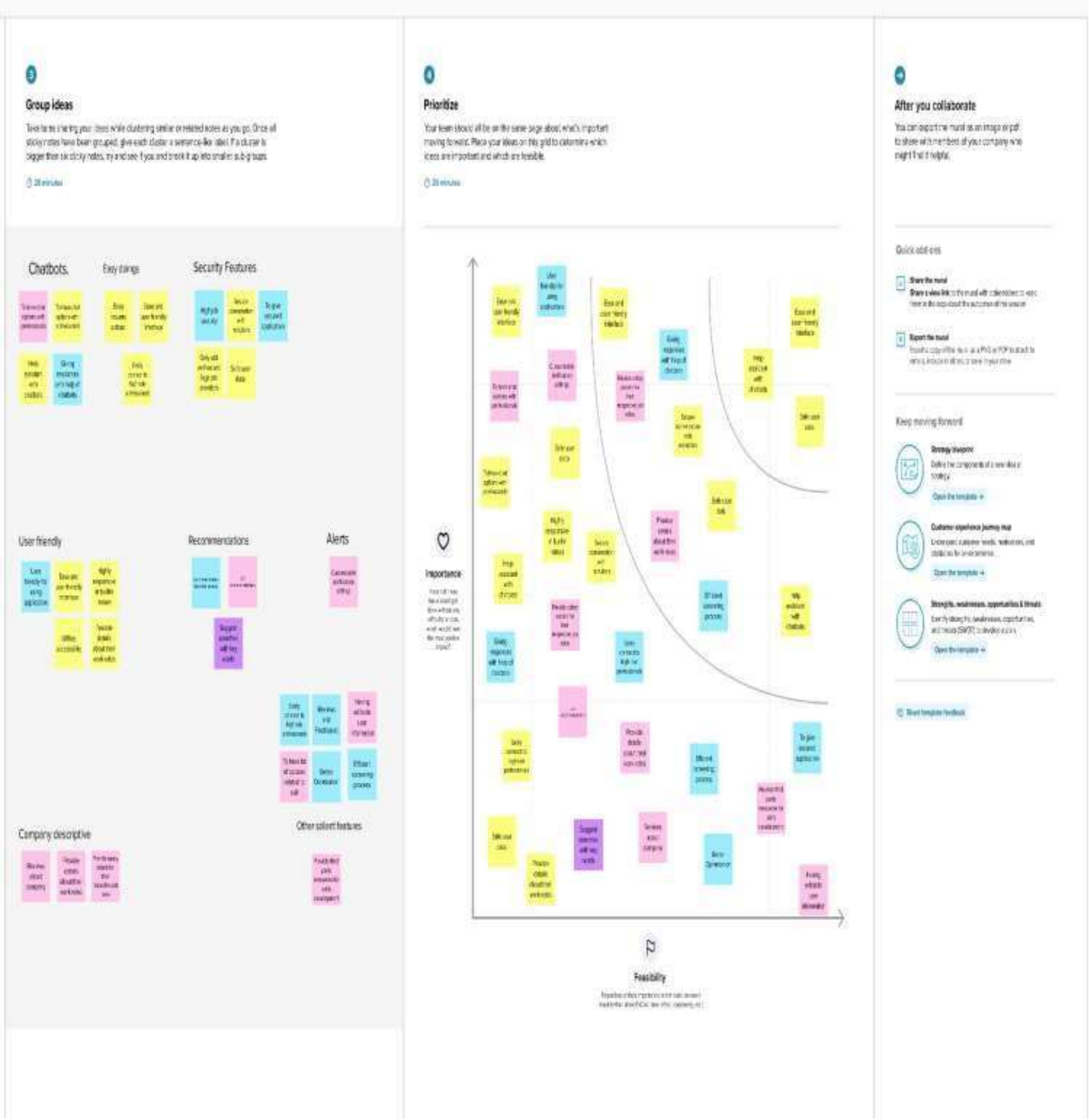
High job security
To give secured application
Giving responses with help of chatbots
User friendly to using application
Efficient screening process
Reviews and Feedbacks
Better [illegible]
Easily connects to high rate professionals
[illegible]

Santhoshkumar

Highly responsive in better network
To gain Happy Customers
Suggest searches with key words
First come - first serve basis
Offline accessibility
[illegible]
Better application tracking
Less wait time
Support with user

Venkatesan

Reviews about company
[illegible]
Provide third party resources for skills development
Provide details about their work roles
Provide salary details for the respective job roles
Customizable notification settings
Having editable user information
To have chat options with professionals
To have list of courses related to skill



Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you can break it up into smaller sub-groups.

20 minutes

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

BRAINSTORMING AND IDEA PRIORITIZATION

3.3 proposed solution

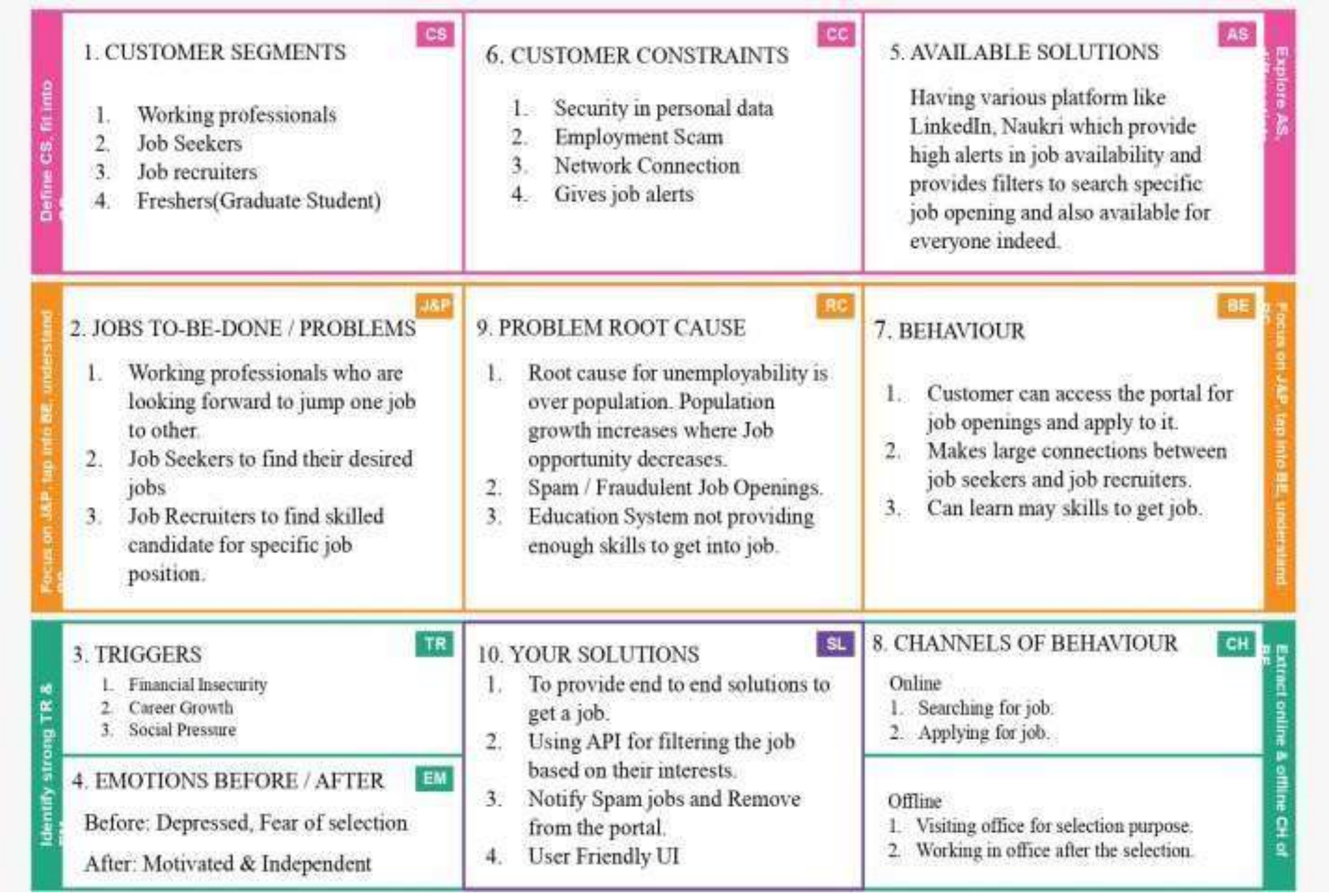
The proposed solution is to improve communication between the project manager and the team, and to give the team more responsibility for their work.

S.No	Parameter	Description
1.	Problem Statement (Problem to be solved)	• When there is an increase in population, there will be a demand for goods and services. For that production manpower needed where automatically hiring happens. • Many industries have various sorts of openings to provide job opportunity, but job seekers are not aware of those openings. And they find it hard to search for a job. • Many users on the internet find it difficult to search through the jobs that are listed in the jobsearching portal. • They might fear fake job offers and not having proper knowledge of the company. • They will apply for all the job openings without doing background verification of the company.
2.	Idea / Solution Description	• Finding the job based on the skills which is updated on the resume of one job seeking personality. • Users get an alert when there is an opening for relevant job they are applied for. • Users can interact with chat-bot. • Fake / Fraudulent job offers are eliminated.
3.	Novelty / Uniqueness	• IBM Watson Assistant is used which acts as an interactive chat-bot in the Portal. • The presence of chat bot removes 24/7 customer support. • It uses Docker to containerize it and IBM cloud for deployment.

4.	Social Impact / Customer Satisfaction	The job and skilled Recommender system is expected to reduce unemployment and improve the skills of job seekers to get a job. The customer satisfaction is measured by getting reviews from the customer after Deployment.
5.	Business Model (Revenue Model)	• Feedback provides two-way communication between end users, which increases trust among users. • When users increase the application is popularized and growth of the application too increases. • Premium features are provided for the users after the application growth.
6.	Scalability of the solution	To provide the best scalability cloud computing is implemented here. The cloud is where the technology gets updated day to day. So, when there is a increases or decreases in IT tools cloud provide the best in it.

3.4 Problem solution fit

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it solves the customer’s problem. The Problem-Solution Fit is an important step towards the Product-Market Fit, but often an underestimated one.



1. CUSTOMER SEGMENTS

1. Working professionals
2. Job Seekers
3. Job recruiters
4. Freshers(Graduate Student)

6. CUSTOMER CONSTRAINTS

1. Security in personal data
2. Employment Scam
3. Network Connection
4. Gives job alerts

5. AVAILABLE SOLUTIONS

Having various platform like LinkedIn, Naukri which provide high alerts in job availability and provides filters to search specific job opening and also available for everyone indeed.

2. JOBS TO-BE-DONE / PROBLEMS

1. Working professionals who are looking forward to jump one job to other.
2. Job Seekers to find their desired jobs
3. Job Recruiters to find skilled candidate for specific job position.

9. PROBLEM ROOT CAUSE

1. Root cause for unemployability is over population. Population growth increases where Job opportunity decreases.
2. Spam / Fraudulent Job Openings.
3. Education System not providing enough skills to get into job.

7. BEHAVIOUR

1. Customer can access the portal for job openings and apply to it.
2. Makes large connections between job seekers and job recruiters.
3. Can learn may skills to get job.

3. TRIGGERS

1. Financial Insecurity
2. Career Growth
3. Social Pressure

4. EMOTIONS BEFORE / AFTER

Before: Depressed, Fear of selection

After: Motivated & Independent

10. YOUR SOLUTIONS

1. To provide end to end solutions to get a job.
2. Using API for filtering the job based on their interests.
3. Notify Spam jobs and Remove from the portal.
4. User Friendly UI

8. CHANNELS OF BEHAVIOUR

Online
1. Searching for job.
2. Applying for job.

Offline
1. Visiting office for selection purpose.
2. Working in office after the selection.

PROBLEM SOLUTION FIT

4. REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements.

Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

The purpose of the Requirements Analysis Phase is to transform the needs and high-level requirements specified in earlier phases into unambiguous (measurable and testable), traceable, complete, consistent, and stakeholder-approved requirements.

4.1 Functional Requirement

Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionalities that define what a system is supposed to accomplish. Behavioral requirements describe all the cases where the system uses the functional requirements, these are captured in use cases.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Mail ID is required

FR-4	User Search	User can search for a job based on their skillset
FR-5	User Profile	Profiles can be edited and updated
FR-6	Chat-Bot	Chat-Bot are provided to help users. If users having doubt regarding job searches, they can reach chat-bot and the bot will answers their queries.

4.2 Non-Functional requirements

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. The plan for implementing functional requirements is detailed in the system design. The plan for implementing non-functional requirements is detailed in the system architecture the system architecture, because they are usually architecturally significant requirements.

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Job seekers are able to search for job openings when there are logged onto the application and can apply for the job with their respected skill set.
NFR-2	**Security**	Separate login for Job Seekers and Job Recruiters.
NFR-3	**Reliability**	Can be accessed without any charges, hence it is opensource application.
NFR-4	**Performance**	Performance-wise it is pretty faster. It gives quick results based on the searching.

NFR-5	**Availability**	Our Skill and Job Recommender application is available widely and also there is wide range of jobs available in the application.
NFR-6	**Scalability**	In order to provide the best scalability, cloud computing is implemented. Response time is also pretty much faster comparatively.

5. PROJECT DESIGN

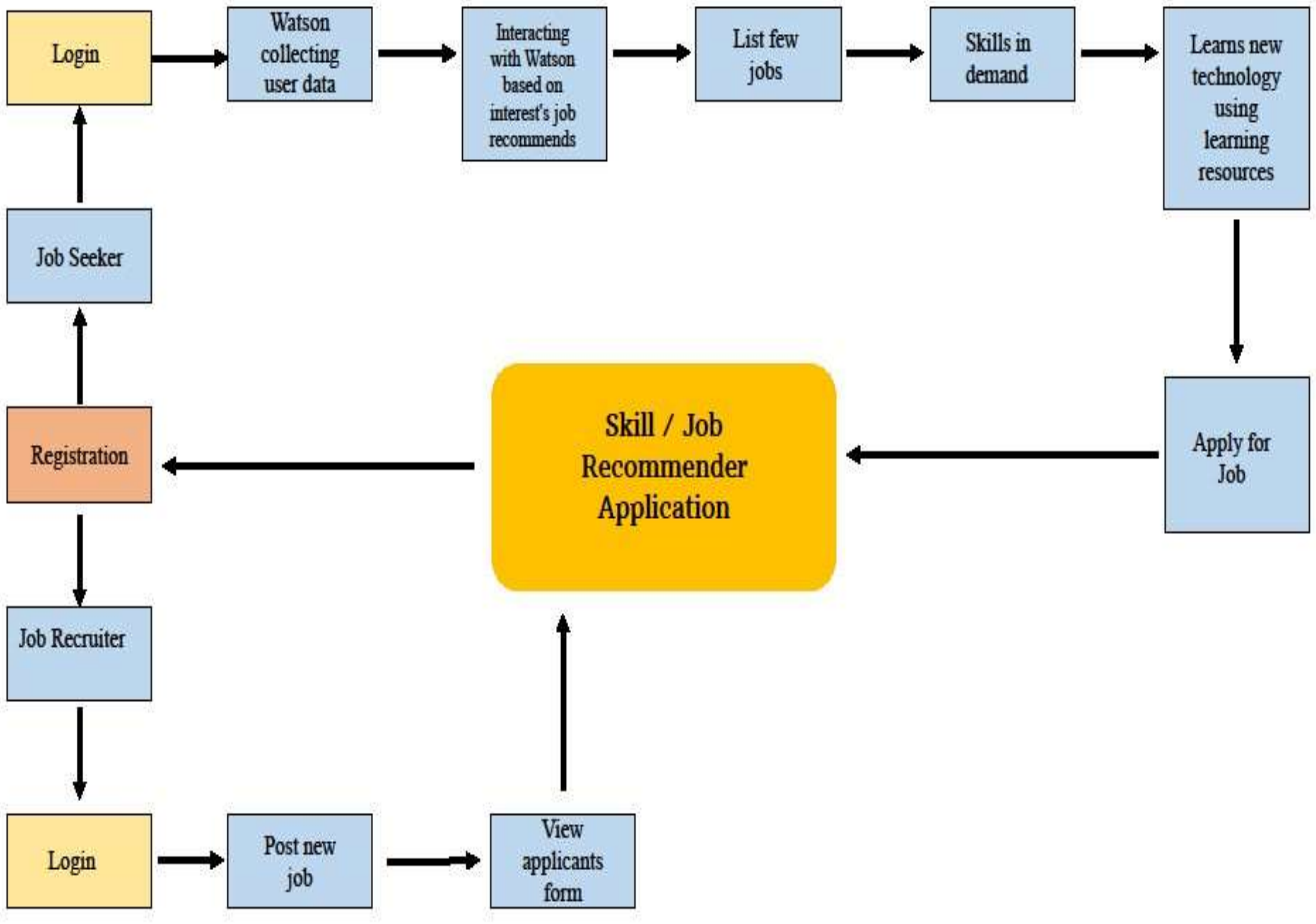
5.1 Data flow diagram

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 can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

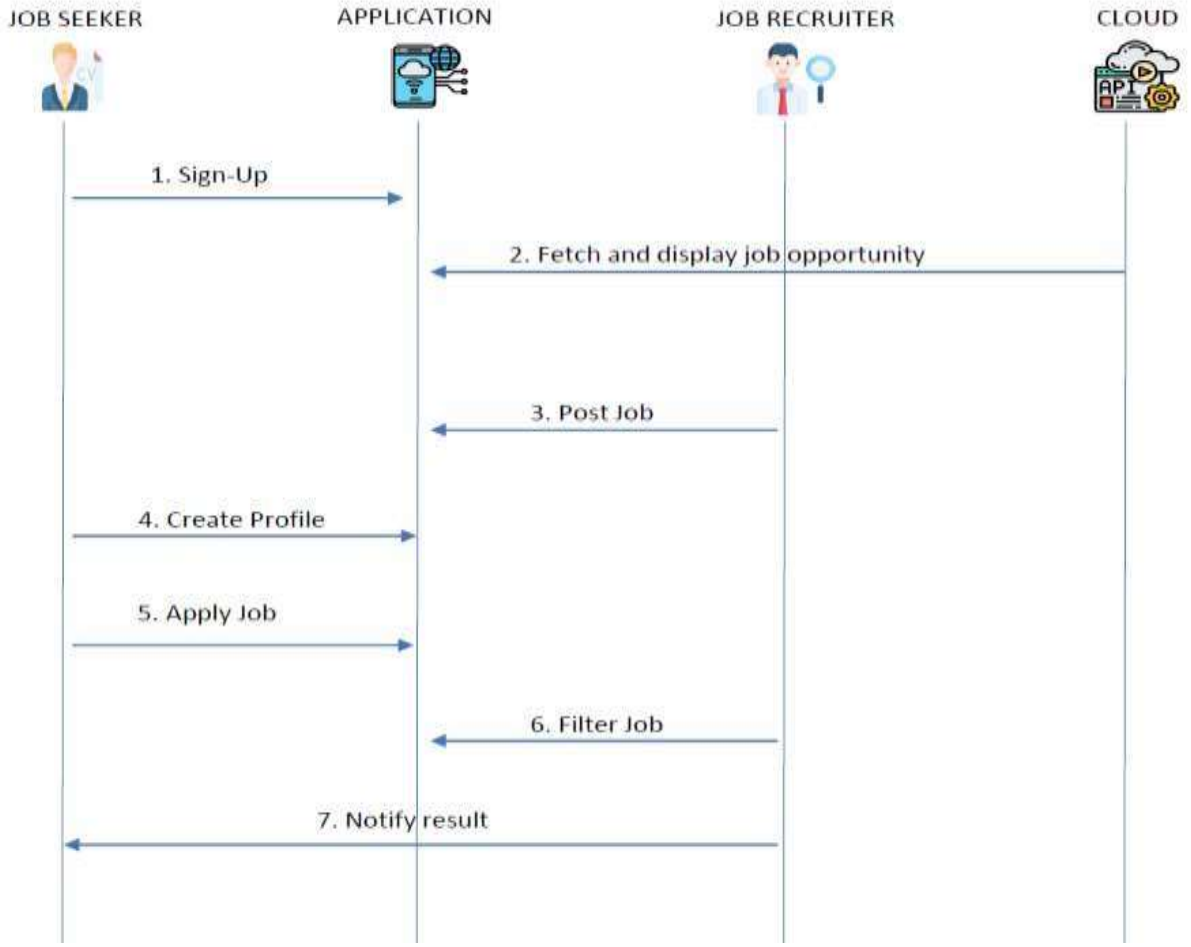
A set of parallel lines shows a place for the collection of data items. A data store indicates that the data is stored which can be used at a later stage or by the other processes in a different order. The data store can have an element or group of elements. The DFD may be used to perform a system or software at any level of abstraction. DFDs may be partitioned into levels that represent increasing information flow and functional detail. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs.

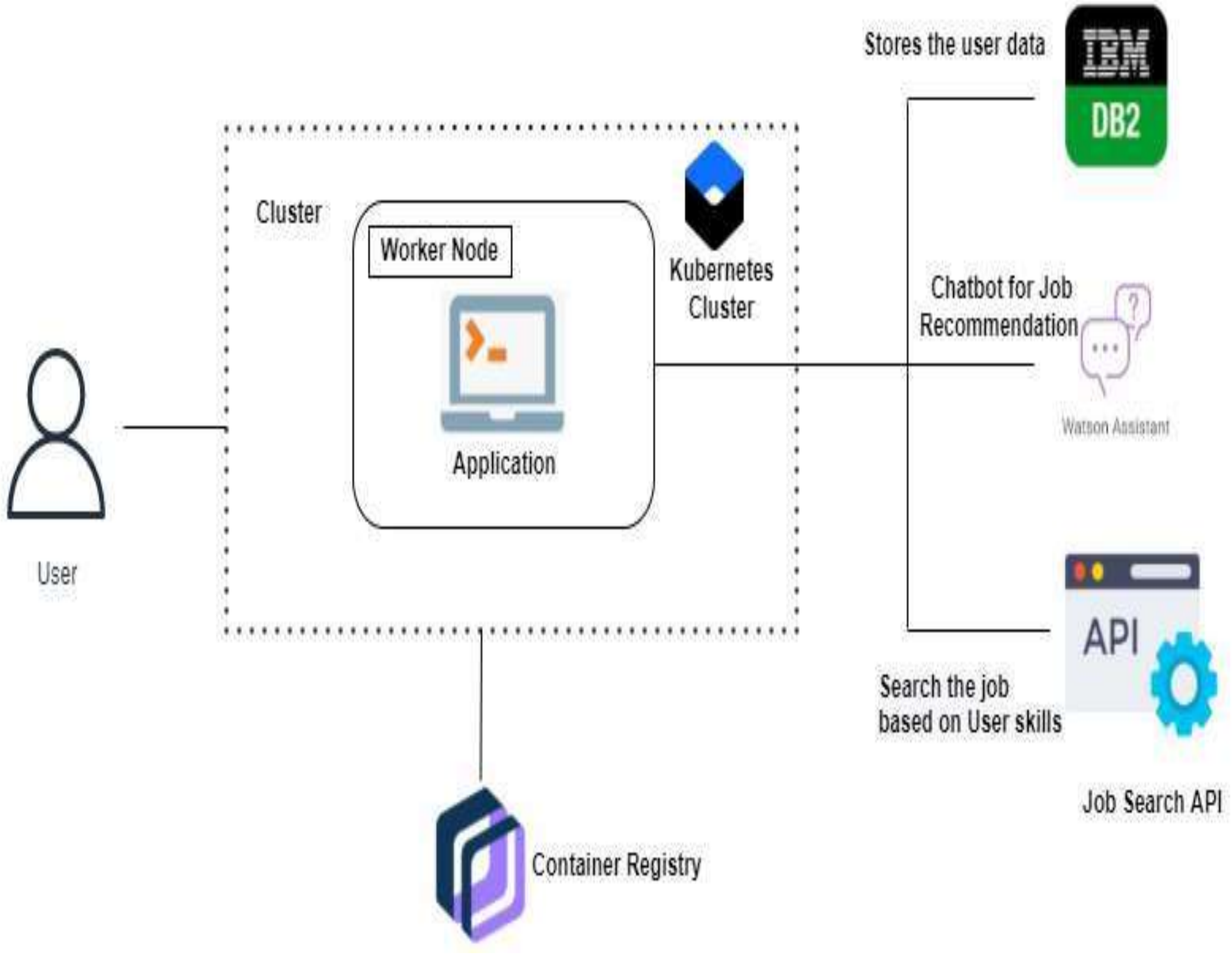

Login
Watson collecting user data
Interacting with Watson based on interest's job recommends
List few jobs
Skills in demand
Learns new technology using learning resources
Job Seeker
Skill / Job Recommender Application
Apply for Job
Registration
Job Recruiter
Login
Post new job
View applicants form

DATA FLOW DIAGRAM

5.2 Solution & Technical Architecture

Solution Architects are most similar to project managers, ensuring that all parties, including stakeholders, are on the same page and moving in the right direction at all stages. Technical architects manage all activities leading to the successful implementation of a new application. They propose a combination of building blocks that provides the best possible fix. This process is very detail-oriented and serves as a connecting piece between enterprise architecture and technical architecture. It also requires a breadth of knowledge in the technical and business inner workings of the company.





TECHNOLOGY STACK

5.3 User Stories

A user story is the smallest unit of work in an agile framework. It’s an end goal, not a feature, expressed from the software user’s perspective.

A user story is an informal, general explanation of a software feature written from the perspective of the end user or customer.

The purpose of a user story is to articulate how a piece of work will deliver a particular value back to the customer. Note that "customers" don't have to be external end users in the traditional sense, they can also be internal customers or colleagues within your organization who depend on your team.

User stories are a few sentences in simple language that outline the desired outcome. They don't go into detail. Requirements are added later, once agreed upon by the team.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	**UI / UX Design**	**USN-1**	**As a user, I need to interact with websites. So, this will be the interface between me and website.**	**High**	**I can interact with website**	**Vasanth, KrishnaRaj**
Sprint-1	**Registration**	**USN-2**	**As a user, I can register for the application by entering my email, password, and confirming my password.**	**High**	**I can register with email**	**Vasanth, Santhosh, Venkatesan**

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1		USN-3	As a user, I will receive confirmation email once I have registered for the application	High	I can receive confirmation mail	Venkatesan, KrishnaRaj
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	Low	I can login to the application by entering login credentials	Vasanth, KrishnaRaj

6. PROJECT PLANNING & SCHEDULING

Planning - Planning pertains to the process of creating a plan of which materials and resources will be required to fulfil incoming and forecasted demand. This step is crucial to ensure that you have enough materials and resource capacity available to produce your orders on time. This component pertains to the 'what' and 'how' of any project: what exactly needs to be achieved and how it will be accomplished.

Scheduling - Scheduling pertains to establishing the timing of the use of specific resources of that organization. In production, scheduling involves developing schedules for workers, equipment, and materials. It reflects on the 'when' of a project, by assigning the appropriate resources to get the production plan completed within a period of time. Creating optimized production schedules ensures that your facility is able to reduce costs, increase productivity, and deliver goods to customers on time.

In order to create accurate and realistic production plans that allow manufacturers to react quickly to changes, it is important to have a production plan that is aligned with the resource and material scheduling process. Having any discrepancy or divergence between the planning and scheduling process creates inefficiencies that can be costly for your business. The bigger the divergence, the larger the cost.

6.1 Sprint planning and estimation

Planning:

In Sprint Planning, the team decides what it will build in the upcoming Sprint and how they will build it. The team commits to the Sprint goal after breaking down user stories into tasks and doing task-level estimation. Sprint Planning is done by the Product Owner, Scrum Master, and the Team. In Scrum, every project is broken into time blocks called sprints, usually 2-4 weeks long. A sprint planning meeting is when the team (including the Scrum Master, Scrum Product Manager, and Scrum Team) meets to determine which backlog items will be handled in the next sprint.

Estimation

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

Product Owner ensures that the prioritized User Stories are clear, can be subjected to estimation, and they are brought to the beginning of the Product Backlog.

As the Scrum Team in total is responsible for the delivery of the product increment, care would be taken to select the User Stories for the Sprint based on the size of the Product Increment and the effort required for the same.

The size of the Product Increment is estimated in terms of User Story Points. Once the size is determined, the effort is estimated by means of the past data, i.e., effort per User Story Point called Productivity.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	UI / UX Design	USN-1	As a user, I need to interact with websites. So, this will be the interface between me and website.	High	I can interact with website	Vasanth, KrishnaRaj
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	High	I can register with email	Vasanth, Santhosh, Venkatesan
Sprint-1		USN-3	As a user, I will receive confirmation email once I have registered for the application	High	I can receive confirmation mail	Venkatesan, KrishnaRaj
Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	Low	I can login to the application by entering login credentials	Vasanth, KrishnaRaj

Sprint-1	Login	USN-4	As a user, I can log into the application by entering email & password	Low	I can login to the application by entering login credentials	Vasanth, KrishnaRaj
Sprint-1	Flask	USN-5	As a user, I can access the website in quick span of time.	High	I can access website quickly	Vasanth, Santhosh, Venkatesan
Sprint-1	Dashboard	USN-6	As a user, I can access various tools and services on the website through the help of dashboard.	High	I can access various tools and services	Vasanth, Santhosh, Venkatesan
			Submission of sprint 1			
Sprint-2	Database	USN-7	As a user, I can store the data am providing in the portal.	High	I can store my data	Vasanth, KrishnaRaj
Sprint-2	User Profile	USN-8	As a user, I can update my profile in the user	Medium	I can modify my details	Santhosh, Venkatesan
			profile section			
Sprint-2	SendGrid Integration	USN-9	As a user, I can get notification through mail services	Medium	I can get mail	Vasanth, KrishnaRaj
Sprint-2	Learning Resources	USN-10	As a user, I can access resources and knowledge which is useful in developing the skills	Medium	I can access learning resources	Venkatesan, KrishnaRaj
Sprint-2	Link to python code	USN-11	As a user, I can access website fast	High	I can access website	Santhosh, Venkatesan, Vasanth
			Submission of sprint 2			
Sprint-3	Cloud Storage	USN-12	As a user, I can store my photos, resumes and much more media that are supported in the webpage.	High	I can store the medias	Vasanth, KrishnaRaj

Sprint-3	Chat-Bot	USN-13	As a user, I can access the chatbot which is very much useful in knowing several things like recent available jobs and jobs opening etc..	High	I can interact with chatbot to solve any queries	Vasanth, Santhosh, Venkatesan
Sprint-3	Integrate to App	USN-14	As a user, I can interact with chatbot after the integration of chat-bot and webpage.	High	Easy access with chat-bot	Vasanth
			Submission of Sprint 3			
Sprint-4	Docker	USN-15	As a user, I can access the webpage in any device.	High	I can access my account in any device	Vasanth, Santhosh, Venkatesan
Sprint-4	Kubernetes	USN-16	As a user, I can access the webpage in any device.	High	I can access my account in any device	Vasanth
Sprint-4	Deployment in cloud	USN-17	As a user, I can access the website any device.	High	I can access my account in any device	Vasanth, KrishnaRaj
Sprint-4	Technical support	USN-18	As a user, I can get a customer care support on the website when am having any queries	Medium	Helps me to solve queries	Vasanth,
Sprint-4	Unit Testing	USN-19	As a user, I can access the website without any interruptions.	High	I can access the website without any interruptions	Vasanth, Krishnaraj, Venkatesan, Santhosh
Sprint-4	Integration Testing	USN-20	As a user, I can access the website without any interruptions.	High	I can access the website without any interruptions	Vasanth, Krishnaraj, Venkatesan, Santhosh
Sprint-4	System Testing	USN-21	As a user, I can access the website without any interruptions.	High	I can access the website without any interruptions	Vasanth, Krishnaraj, Venkatesan, Santhosh

Sprint-4	**Correction**	**USN-22**	**As a user, I can access the website without any interruptions.**	**High**	**I can access the website without any interruptions**	**Vasanth, Krishnaraj, Venkatesan, Santhosh**
Sprint-4	**Acceptance Testing**	**USN-23**	**As a user, I can access the website without any interruptions.**	**High**	**I can access the website without any interruptions**	**Vasanth, Krishnaraj, Venkatesan, Santhosh**
			Submission of Sprint 4			

PLANNING & ESTIMATION

6.2 Sprint delivery schedule

Since sprints take place over a fixed period of time, it’s critical to avoid wasting time during planning and development. And this is precisely where sprint scheduling enters the equation.

In case you’re unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It’s one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

Teams often run into trouble when they create more than a few schedules. This can create conflict and derail projects midway through their cycles. To ensure things stay on track, one schedule makes sense.

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**	**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:

We have a 6-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)**

AV = sprint duration / velocity

AV = 20 / 6

AV = 3.33 SPRINT

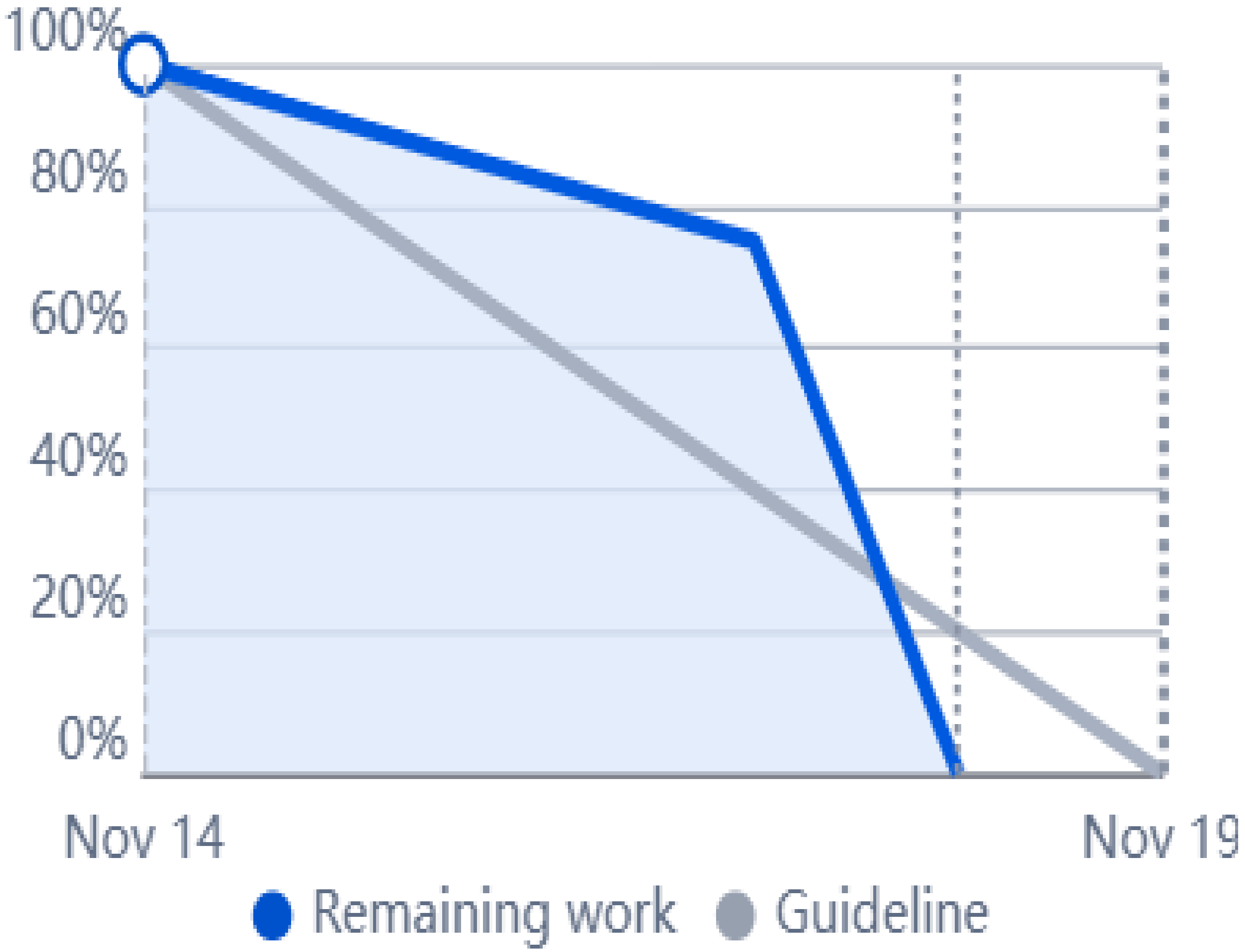
DELIVERY SCHEDULE

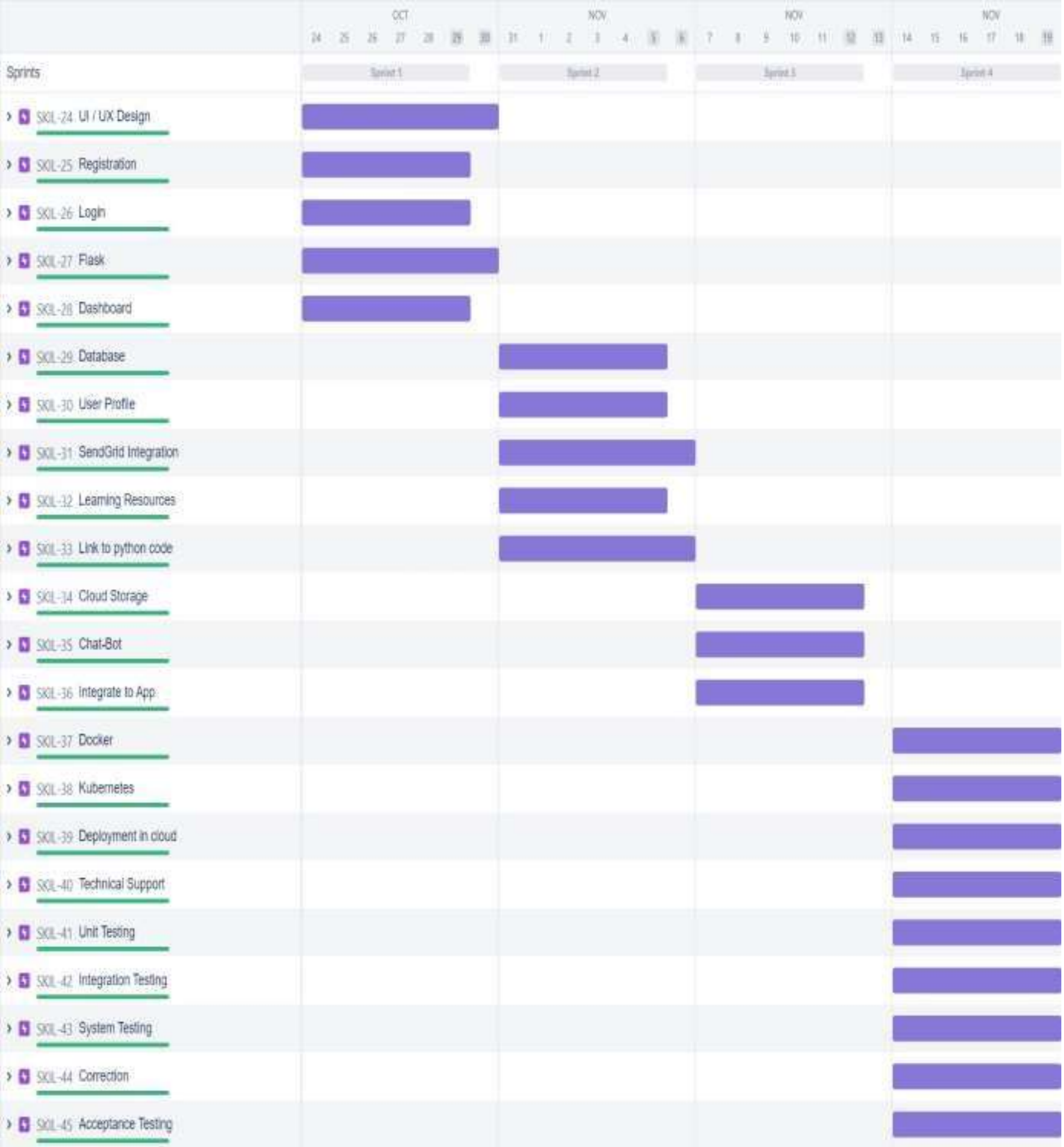
6.3 Reports from JIRA

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.

Typically, in a burn down chart, the outstanding work is often on the vertical axis, with time along the horizontal. It is useful for predicting when all of the work will be completed. In the Daily Scrum the Development Team updates the Sprint Burn Down and plots the remaining work of the day.

Burndown Chart(Sprint 4):


100%
80%
60%
40%
20%
0%
Nov 14
Nov 19
● Remaining work
● Guideline



SKIL-24 UI / UX Design
SKIL-25 Registration
SKIL-26 Login
SKIL-27 Flask
SKIL-28 Dashboard
SKIL-29 Database
SKIL-30 User Profile
SKIL-31 SendGrid Integration
SKIL-32 Learning Resources
SKIL-33 Link to python code
SKIL-34 Cloud Storage
SKIL-35 Chat-Bot
SKIL-36 Integrate to App
SKIL-37 Docker
SKIL-38 Kubernetes
SKIL-39 Deployment in cloud
SKIL-40 Technical Support
SKIL-41 Unit Testing
SKIL-42 Integration Testing
SKIL-43 System Testing
SKIL-44 Correction
SKIL-45 Acceptance Testing

7. CODING AND SOLUTIONING

7.1 Feature 1

```
import configparser

import sendgrid
from sendgrid.helpers.mail import Mail

config = configparser.ConfigParser() import
base64

config.read('mail.env')
APIKEY = config.get('API', 'APIKEY') api =
sendgrid.SendGridAPIClient(APIKEY)
FROM_EMAIL =
config.get('API','FROM_EMAIL') def
sendemail(user,content):
    TO_EMAIL = user

    # if type == 'Account_creation':
    mail =
Mail(from_email=FROM_EMAIL,to_emails=TO_EMAIL,subject='Hey there!
We heard from you!',html_content=f'<strong>{content}</strong>')     # if type ==
'complaint_creation':
    #     mail =
Mail(from_email=FROM_EMAIL,to_emails=TO_EMAIL,subject='Complaint
Created successfully',html_content='<strong>Compliant created
Successfully</strong>')
response = api.send(mail)
print(response.status_code)
    print(response.headers)
```

Feature 2

```
{% extends 'base.html' %}

{%  block head %}
    <title>Features - Dream Team Recruitment™</title>
<link type="image/png" sizes="16x16" rel="icon"
href="static/img/job_logo.png">
    <link rel="stylesheet" href="static/css/bootstrap.min.css">
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Raleway:300italic,400italic,600italic,
700italic,800italic,400,300,600,700,800&amp;display=swap">
    {% endblock %}



    {% block content %}
    <script>
        window.watsonAssistantChatOptions = {           integrationID: "a93c8bfc-
cda0-49cd-a3d0-b68581f7ef06", // The ID of this integration.           region: "eu-
gb", // The region your integration is hosted in.
          serviceInstanceID: "1ab40042-c8f9-455b-b469-fa7435694735", // 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>
```

Coding:

App.py

```
import ibm_db
from flask import Flask, flash, redirect, render_template, request, url_for
from flask_mail import Mail, Message from sendgrid import
SendGridAPIClient
from sendgrid.helpers.mail import Mail

from sendmailer import *
import pandas as pd

app = Flask(__name__)



# app.secret_key="
# configure the mail settings

SENDGRID_API_KEY =
"SG.bnGBaY6cSGeU106QGq_H5Q.YhqfT29UYDRV9yWp3Rfn73LQykmE455
Zckt_qyJSR2U"
app.config['SECRET_KEY'] = 'top-secret!'
app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
app.config['MAIL_PORT'] = 587
app.config['MAIL_USE_TLS'] = True
app.config['MAIL_USERNAME'] = 'apikey'
# app.config['MAIL_PASSWORD'] = os.environ.get('SENDGRID_API_KEY')
# app.config['MAIL_DEFAULT_SENDER'] =
os.environ.get('vasanthias52@gmail.com') mail
= Mail(app)

conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-
46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=3124
9;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=pfd0
3782;PWD=dnhtwrcfkZlhkhAO",",")  # type: ignore print(conn)
print("connection successful...")
```

```
@app.route('/sendgrid') def sendgrid():
return render_template('sendgrid.html')



@app.route('/loginpage', methods=['POST','GET'])
def loginpage():     if request.method == 'POST':
email = request.form['email']         password =
request.form['password']         if not email or not
password:
         return render_template('login.html',error='Please fill all fields')

      query = "SELECT * FROM USERS WHERE EMAIL=? AND PASSWORD=?"
      stmt = ibm_db.prepare(conn, query) # type:ignore
ibm_db.bind_param(stmt,1,email) # type:ignore
ibm_db.bind_param(stmt,2,password) # type:ignore
ibm_db.execute(stmt) # type:ignore         isUser =
ibm_db.fetch_assoc(stmt) # type:ignore
print(isUser,password)

      if not isUser:
         return render_template('login.html',error='Invalid Credentials')

      return redirect(url_for('home'))

   return render_template('login.html',name='Home')


@app.route('/signup') def registration():
return render_template('signup.html')


@app.route('/signup', methods=['POST','GET']) def
signup():
   if request.method == 'POST':
name = request.form['name']
email = request.form['email']
phone = request.form['phone']
      password = request.form['password']
      sql ="INSERT INTO USERS VALUES (?,?,?,?)"
stmt = ibm_db.prepare(conn,sql)  # type: ignore
ibm_db.bind_param(stmt, 1, name)  # type: ignore
ibm_db.bind_param(stmt, 2, email)  # type: ignore
ibm_db.bind_param(stmt, 3, phone)  # type: ignore
ibm_db.bind_param(stmt, 4, password)  # type: ignore
ibm_db.execute(stmt)  # type: ignore
sendemail(email,")        return redirect(url_for('home'))

   return render_template('signup.html')



@app.route('/') @app.route('/login')
def login():     return
render_template('login.html')






@app.route('/stats') def stats():
return render_template('stats.html')

@app.route('/contacts') def
requester():
   return render_template('contacts.html')




@app.route('/tech', methods=['POST']) def
by_tech():
   jobs = pd.read_csv('jobs.csv')
input = request.get_json()
tech_name = input['techName']
   page = int(input['page']) if input['page'] else 0
sorting = input['sorting'] if input['sorting'] else 0

   filtered_jobs = jobs.loc[jobs['Tech Stack'].str.contains(tech_name, na=False)]

   if sorting:
        filtered_jobs = filtered_jobs.sort_values(by=[sorting])

    return              filtered_jobs[page*10:page*10+10].drop(['Unnamed:              0'],
axis=1).to_json(orient='records')



@app.route('/location', methods=['POST']) def
by_location():     jobs = pd.read_csv('jobs.csv')
input = request.get_json()     location =
input['location']     page = int(input['page']) if
input['page'] else 0     sorting = input['sorting'] if
input['sorting'] else 0

    filtered_jobs  =  jobs.loc[jobs['Location']  ==  location]
if sorting:
        filtered_jobs = filtered_jobs.sort_values(by=[sorting])

    return              filtered_jobs[page*10:page*10+10].drop(['Unnamed:              0'],
axis=1).to_json(orient='records')

@app.route('/forgot') def
reques():
    return render_template('forgotten-password.html')

@app.route('/forgot',methods=['POST','GET'])
def forgot():     if request.method == 'POST':
email = request.form['email']
        query = "SELECT * FROM USERS WHERE EMAIL=?"
stmt = ibm_db.prepare(conn, query) # type:ignore
ibm_db.bind_param(stmt,1,email)        ibm_db.execute(stmt)
# type:ignore
        isUser = ibm_db.fetch_assoc(stmt) # type:ignore
        # print(isUser,password)
        print(isUser)
print(stmt)
        sendemail(email,'We have recieved your email! from your email address to
reset the password, we will send you a link to reset your password')        return
render_template('login.html')
    return render_template('forgotten-password.html')
```

```
@app.route('/features') def
features():
   return render_template('features.html')


@app.route('/home') def home():
return render_template('index.html')

# @app.route('/') #
def home():
#    return render_template('index.html')

@app.route('/contacts' ,methods=['POST'])
def contacts():    email =
request.form['email']
   sendemail(email,'We have recieved your email!')
return render_template('contacts.html')


@app.route('/logout') def
logout():

   session.pop('email', None) # type:ignore
   return redirect(url_for('login'))


if __name__=='__main__':
app.run(debug=True)
```

Output:

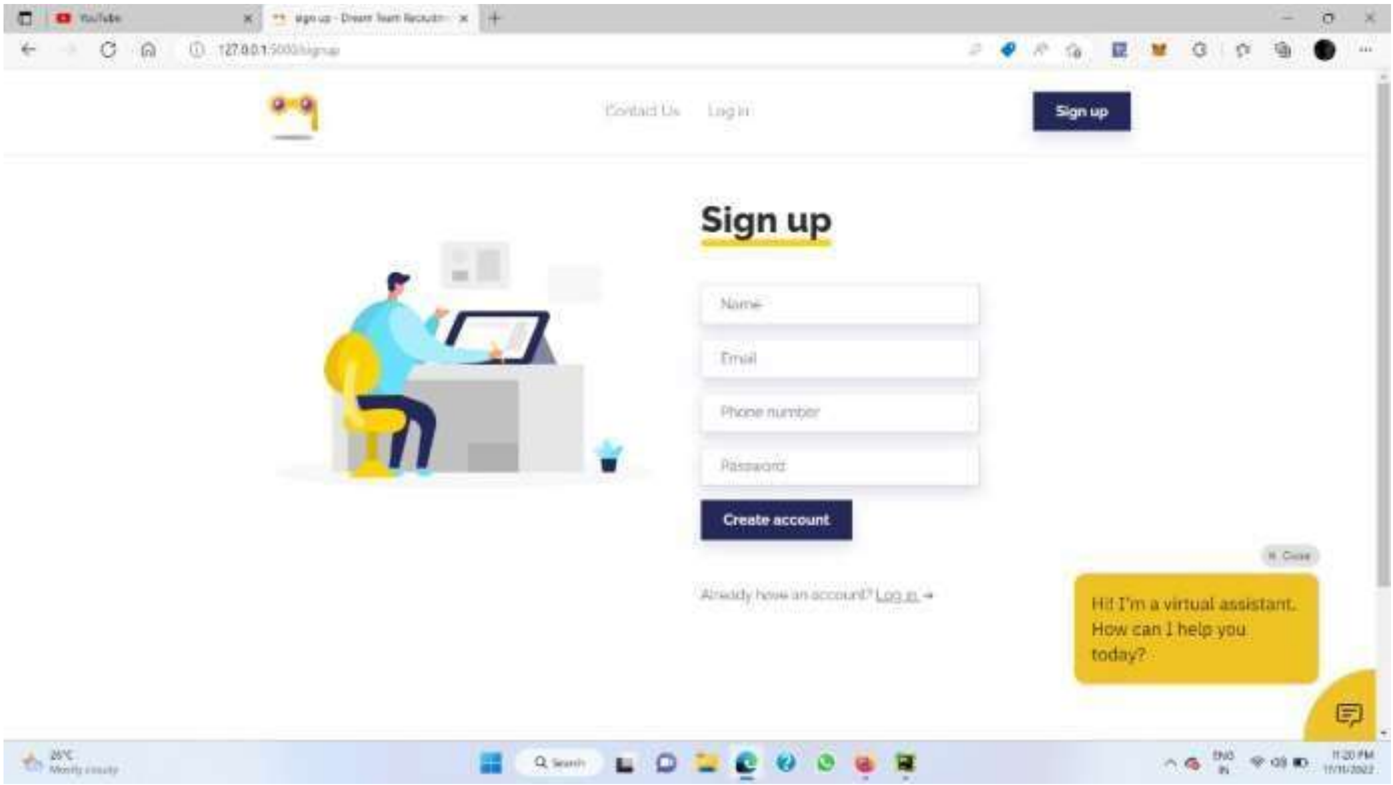
```
connection successful...
* Serving Flask app 'app' (lazy loading)
* Environment: production
         WARNING: This is a development server. Do not use it in a production
deployment.
  Use a production WSGI server instead.
```

* Debug mode: on

WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.

* Running on http://127.0.0.1:5000

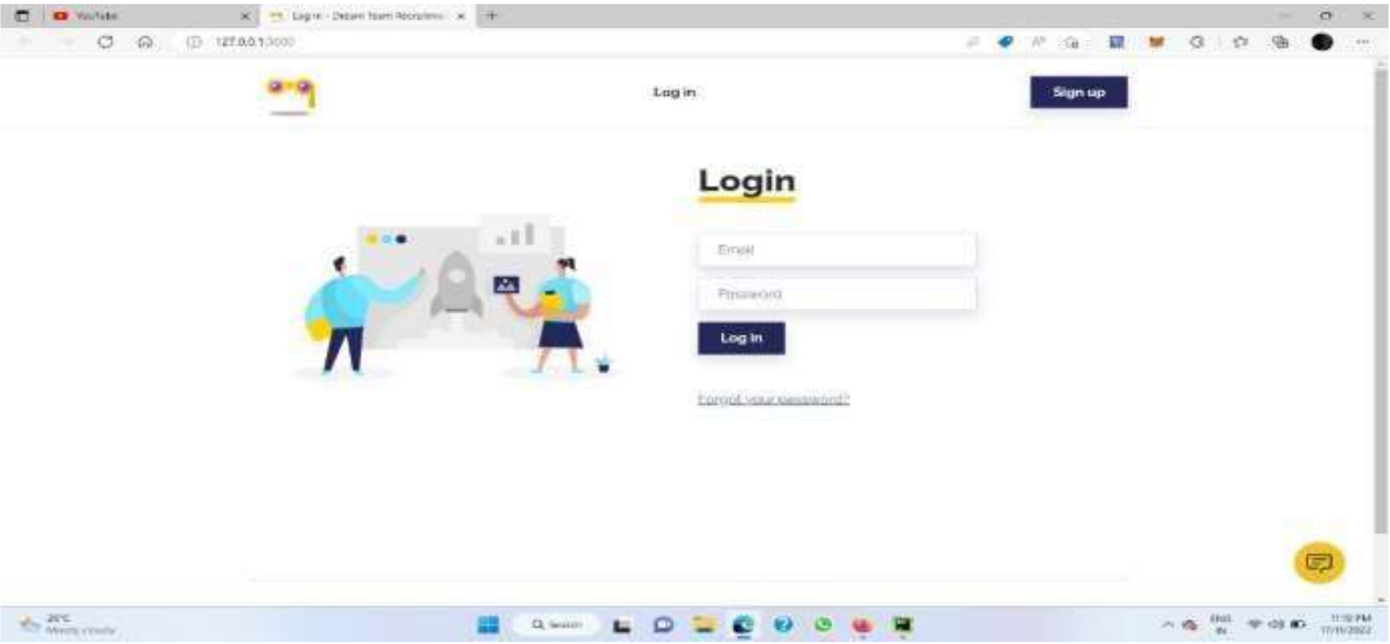
OUTPUT IMAGES:



Sign up

Already have an account? Log in →

Hi! I'm a virtual assistant. How can I help you today?



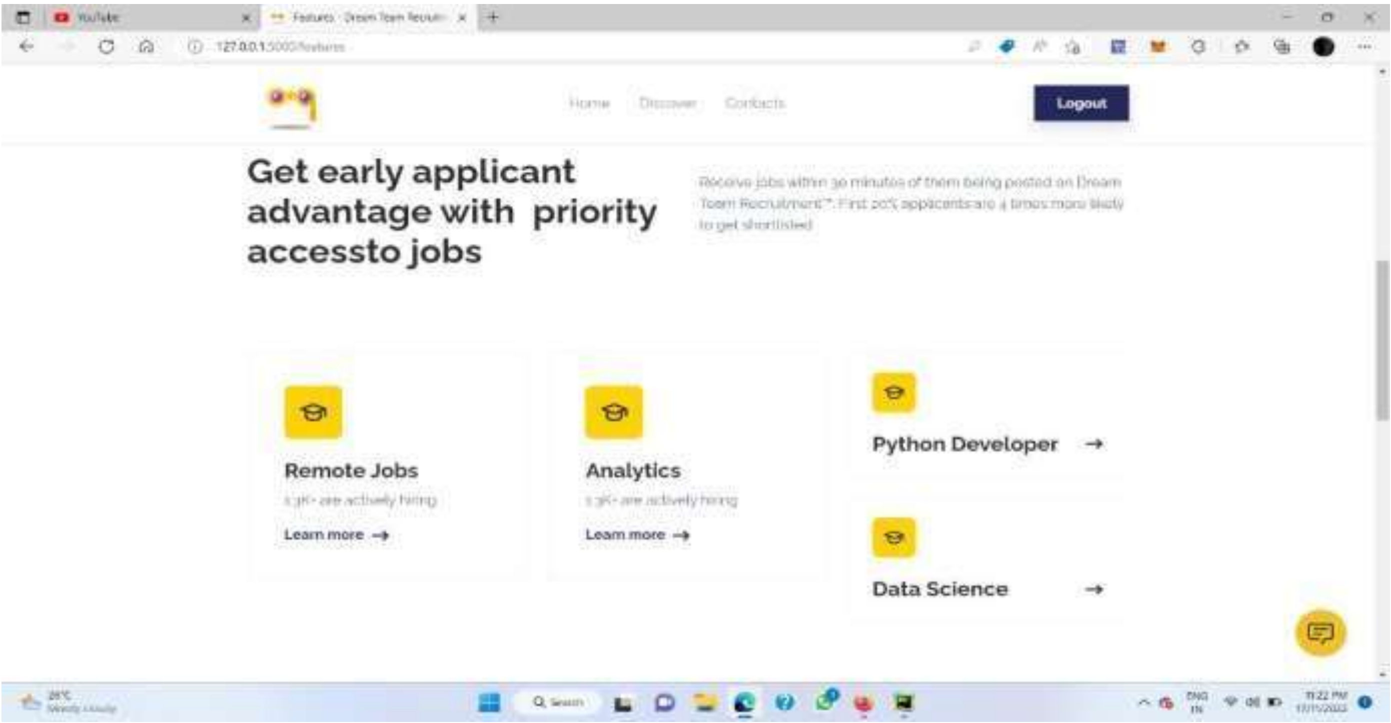
Login

Email

Password

Log in

Forgot your password?



Get early applicant advantage with priority accessto jobs

Receive jobs within 30 minutes of them being posted on Dream Team Recruitment™. First 20% applicants are 4 times more likely to get shortlisted

Remote Jobs

1.3K+ are actively hiring

Learn more →

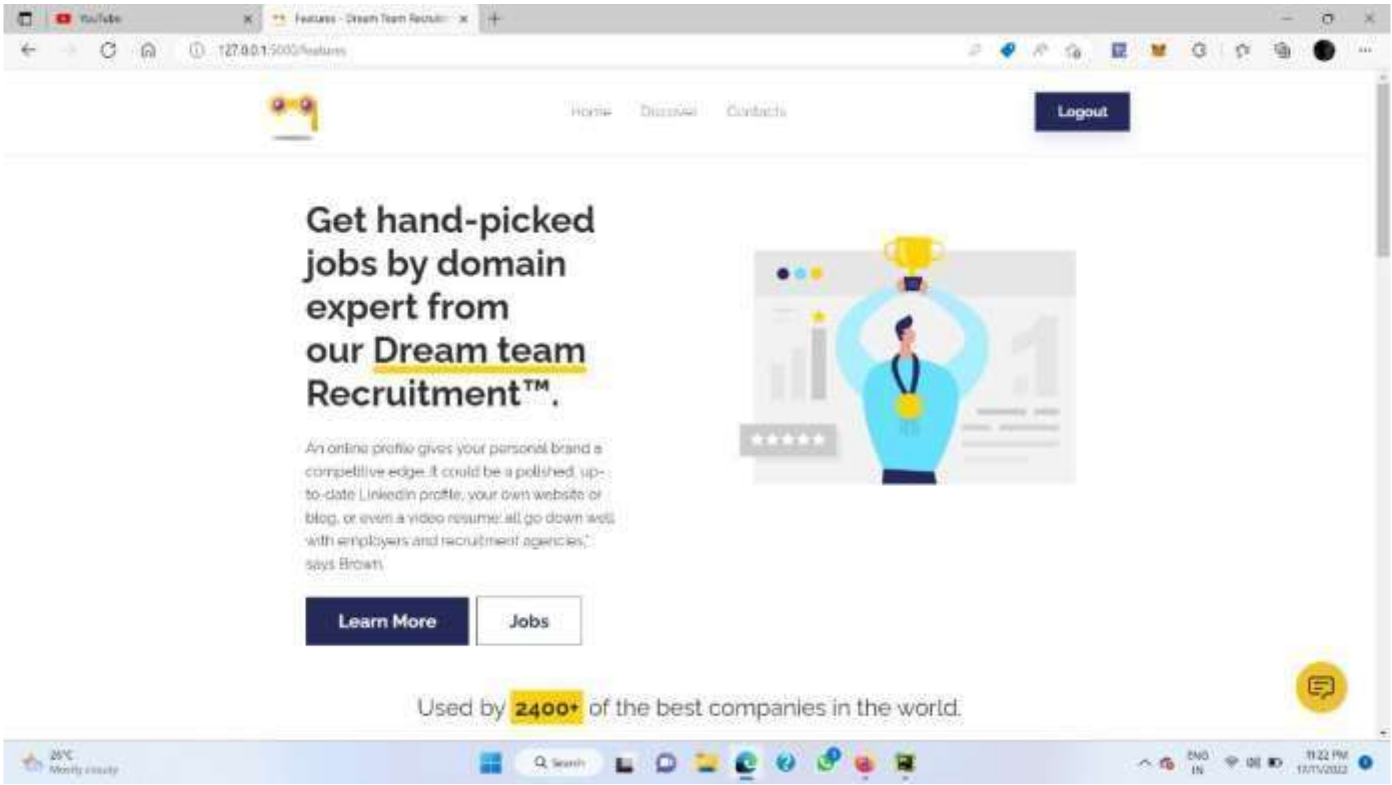
Analytics

1.3K+ are actively hiring

Learn more →

Python Developer →

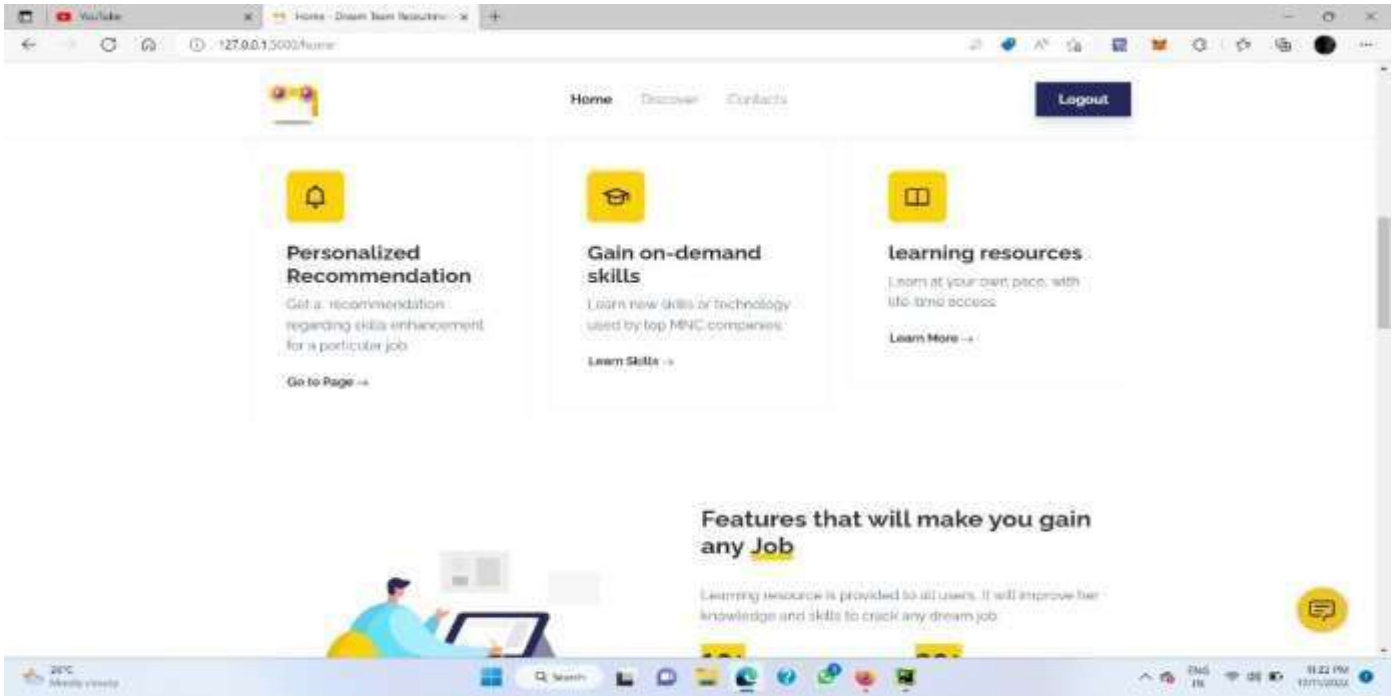
Data Science →



Get hand-picked jobs by domain expert from our Dream team Recruitment™.

An online profile gives your personal brand a competitive edge. It could be a polished, up-to-date LinkedIn profile, your own website or blog, or even a video resume: all go down well with employers and recruitment agencies," says Brown.

Used by 2400+ of the best companies in the world.



Personalized Recommendation

Get a recommendation regarding skills enhancement for a particular job

Gain on-demand skills

Learn new skills or technology used by top MNC companies

learning resources

Learn at your own pace, with life time access

Features that will make you gain any Job

Learning resource is provided to all users. It will improve her knowledge and skills to crack any dream job

8. TESTING

Software testing is used to assess the quality of the product. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation.

8.1 Test Cases:

Testcase1: Does the flask application is perfectly created and in works in very good condition?

Testcase2: Does the Send-Grid integration is working correctly?

Testcase3: Does the db2 is perfectly connected to the application? **Testcase4:** Can the chat-bot which is created using Watson assistant is recommending correctly the job for the end users?

Testcase 5: Whether the application is working correctly without any interruptions?

8.2 User Acceptance Testing

User acceptance testing is a type of testing that is used to determine whether or not a software system is suitable for use by end users. It is the process of verifying that a system meets the requirements of the user and that the user is able to use the system for its intended purpose. User acceptance testing (UAT) is a process of verifying that a system meets the needs of the end users and that they are able to use it. This can be done through a variety of methods, such as interviews, surveys, or observation. UAT is important in water quality analysis and prediction because it helps ensure that the system being developed will be useful to those who will be using it. By testing with act users, developers can get feedback on the system and make sure it is meeting the needs of the users.

9. RESULTS

9.1 Performance Metrics

a) Implementation of web application:

To create the web application to interact with the users. The users here is commonly job seeker and job provider. Login, Signup, Job searching have separate pages where we can access into different work functions.

b) SendGrid Integration:

The flask application that we created is to get integrated with sendgrid which provide the e-mail interface for communication purpose.

c) Developing chatbot:

To develop a chat-bot so that, that can be very interactive to the users who are using the application and to recommend the jobs based on the job seekers interests.

d) Deployment of Application:

Finally the developed application is to deployed in the cloud.

1 .Accuracy

The accuracy metric is one of the simplest Classification metrics to implement, and it can be determined as the number of correct predictions to the total number of predictions.

To implement an accuracy metric, we can compare ground truth and predicted value in a loop.

10. ADVANTAGES AND DISADVANTAGES

Advantages:

- The main advantage of our application is that there is a direct one way communication between the job seeker and the job recruiter.
- There is a chat-bot which gives the directions to the users what to do and not to do and also it recommends the jobs based on the job seekers interests.
- The application is an open source one which doesn’t asks for the money.

Disadvantage:

- One disadvantage of the application is that it is not a full paced one.
- Another disadvantage of skill / job recommender application is that it is unaware of machine language stack. No AI is implemented here.
- Skill / Job Recommender Application is used in many domains despite of professionals.

11. CONCLUSION

In this work, we have presented our proposal for the automatic recommendation of job offers. Our goal here is being able to build methods being able to deliver appropriate job offers to those job seekers that could be potentially interested on them. To do that, we have based our research efforts on two wellknown classification methods: random forests (RF) and support vector machines (SVM). Our empirical evaluation shows us interesting facts. For example, RF are more likely to be interpreted although they do no present a particularly good performance in relation to SVM. On the other hand, SVM are more accurate, although they work with a model being much harder to interpret by human. What it is clear is, that in both cases, we have shown that these two methods are quite appropriate for accurately working in the context of automatic job recommendation.

There is unemployment only because of lack of skill set in their domain or fear of missing out on a job. When a job seeker is afraid of getting the desired job, he might losses the job which he really deserved for it. The main purpose of the job recommender application is to provide job opportunities for each and every single person. The only thing the job seeker wants to do is just to approach the application and apply for the job. He will be provided with the login credentials with the confirmation email. There he can find numerous job opportunities. He will be guided with the in-built chat bot, which guides the job seekers to apply for the job and recommends the availability of jobs based on their interest. The chat bot is built with IBM Watson Assistant that is very much helpful in collecting the job seeker interests and also guides them to apply for it.

12. FUTURE SCOPE

As future work, we propose to design novel computational methods being able to process the textual description from the job offers. At that point, we were using just the quantitative information that is advertised. However, we think that the way an offer is written can help attracting potential candidates as well, maybe new methods for natural language processing using neural networks could help in this task. We also would like to explore the possibilities to work with expert knowledge via kernel mapping in the case of SVM as we mentioned earlier. Finally, it is also necessary to study how to integrate this technology with existing web information systems so that these two methods can be put into operation by the industry.

.