

# **Skill / Job Recommender Application**

**Team ID-** PNT2022TMID53102

**Batch-** B7-1A3E

## **PROJECT REPORT**

### **1. INTRODUCTION**

Having lots of skills but wondering which job will best suit you? Don't need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.

#### **1.1 PROJECT OVERVIEW**

There has been a sudden boom in the technical industry and an increase in the number of good startups. Keeping track of various appropriate job openings in top industry names has become increasingly troublesome. This leads to deadlines and hence important opportunities being missed. Through this research paper, the aim is to automate this process to eliminate this problem. To achieve this, IBM cloud services like db2, Watson assistant, cluster, kubernetes have been used. A hybrid system of Content-Based Filtering and Collaborative Filtering is implemented to recommend these jobs. The intention is to aggregate and recommend appropriate jobs to job seekers, especially in the engineering domain. The entire process of accessing numerous company websites hoping to find a relevant job opening listed on their career portals is simplified. The proposed recommendation system is tested on an array of test cases with a fully functioning user interface in the form of a web application. It has shown satisfactory results, outperforming the existing systems. It thus testifies to the agenda of quality over quantity.

## 1.2 PURPOSE

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.

## 2. LITERATURE SURVEY

### **Abstract**

In this project, the job recommendation systems aim to minimize information overload by helping user's in searching desired job information. Faced with this scenario, we investigate the use of cloud factors able to have a positive influence on generating recommendations. Thus, we present a new, simple model based on cloud features which is associated with the content-based technique of job recommendation. The practical applicability of data storage environments in the cloud provides the best use of cloud resources and meets user's preferences.

In this project, We

- Made publicly accessible a new dataset formed by a set of job seekers profiles and a set of job vacancies amassed from distinctive job search engine sites
- Put ahead the proposal of a framework for job suggestion based totally on expert competencies of job seekers
- Carried out an assessment to quantify empirically the advice competencies of two state-of-the-art methods, thinking about special configurations, inside the proposed framework.

### **Related Projects**

#### **1. A survey of job recommender systems**

Shaha T. Al-Otaibi<sup>1</sup> \* and Mourad Ykhlef<sup>2</sup>

The Internet-based recruiting platforms become a primary recruitment channel in most companies. While such platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the opportunity of recruiting. The recommender system technology aims to help users in finding items that match their personnel interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. In order to improve the e-recruiting functionality, many recommender system approaches have been proposed. This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

**Key words:** Recommender systems, collaborative filtering, content-based filtering, hybrid approach, machine learning, e-recruiting, similarity measure.

2. M. Diaby and E. Viennet, "**Taxonomy-based job recommender systems on Facebook and LinkedIn profiles**," 2014 IEEE Eighth International Conference on Research Challenges in Information Science (RCIS), 2014, pp. 1-6, doi: 10.1109/RCIS.2014.6861048.

This paper presents taxonomy-based recommender systems that propose relevant jobs to Facebook and LinkedIn users; they are being developed by Work4, a San Francisco-based software company and the Global Leader in Social and Mobile Recruiting that offers Facebook recruitment solutions; to use its applications, Facebook or LinkedIn users explicitly grant access to some parts of their data, and they are presented with the jobs whose descriptions are matching their profiles the most. In this paper, we use the O\*NET-SOC taxonomy, a taxonomy that defines the set of occupations across the world of work, to develop a new taxonomy-based vector model for social network users and job descriptions suited to the task of job recommendation; we propose two similarity functions based on the AND and OR fuzzy logic's operators, suited to the proposed vector model. We compare the performance of our proposed vector model to the TF-IDF model using our proposed similarity functions and the classic heuristic measures; the results show that the taxonomy-based vector model outperforms the TF-IDF model. We then use SVMs (Support Vector Machines) with a mechanism to handle unbalanced datasets, to learn

similarity functions from our data; the learnt models yield better results than heuristic similarity measures. The comparison of our methods to two methods of the literature (a matrix factorization method and the Collaborative Topic Regression) shows that our best method yields better results than those two methods in terms of AUC. The proposed taxonomy-based vector model leads to an efficient dimensionality reduction method in the task of job recommendation.

### **3. CaPaR: A Career Path Recommendation Framework**

Authors: Magdalini Eirinaki, Bharat Patel, Varun Kakuste

Published in: 2017 IEEE Third International Conference on Big Data Computing Service and Applications (BigDataService)

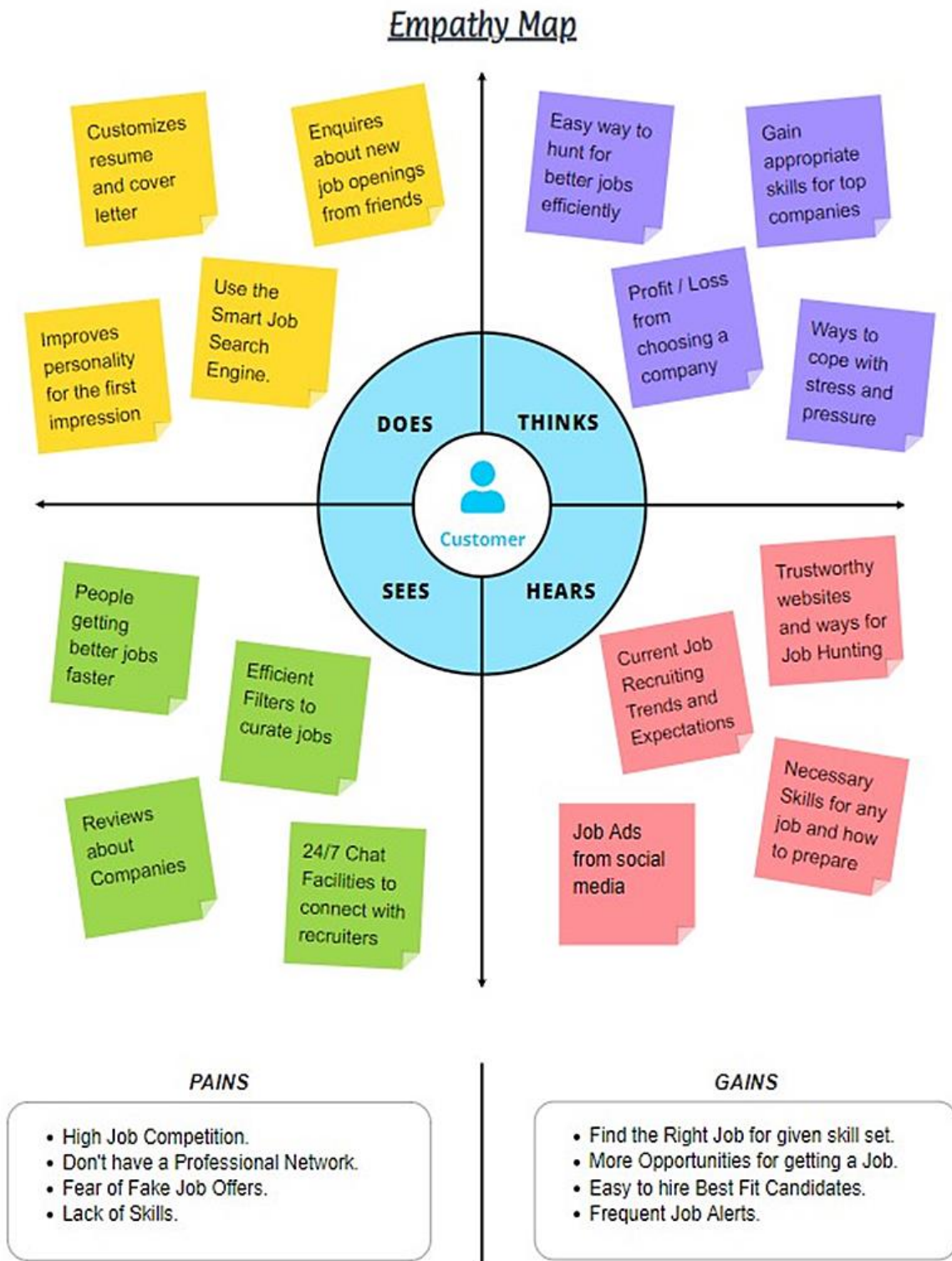
Existing job suggestion structures only reflect on consideration on the user's subject of pastime and omit the user's profile and skills, which should result in extra relevant career guidelines for users. CaPaR, a Career Path Recommendation framework, is proposed in this paper to address such shortcomings. The gadget scans the user's profile and resume, identifies the candidate's key skills, and generates customized job recommendations using textual content mining and collaborative filtering techniques. Furthermore, the device suggests to student's extra competencies needed for related job openings, as nicely as learning assets for each skill. As a result, the gadget not only permits its customers to explore big quantities of information, but additionally to enlarge their portfolio and resume in order to enhance their careers.

### **3. IDEATION AND PROPOSED SOLUTION:**

#### **3.1 Empathy Map**

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to

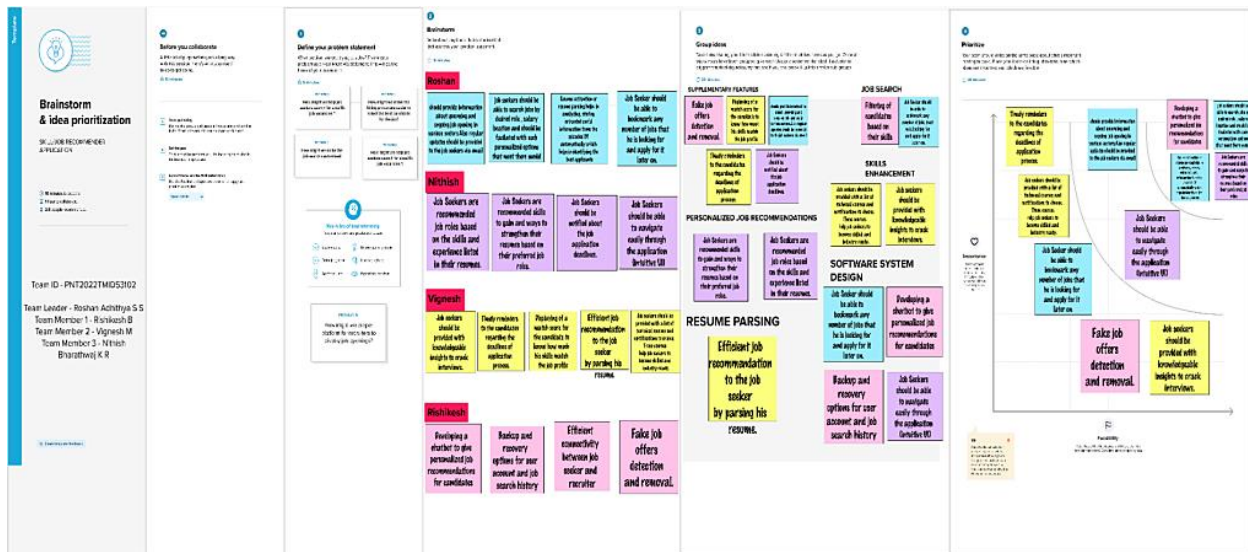
- Create a shared understanding of user needs, and
- Aid in decision making



### 3.2 IDEATION AND BRAINSTROMING:

## Brainstorm & Idea Prioritization Template:

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. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.



### 3.3 PROPOSED SOLUTION

Having lots of skills but wondering which job will best suit you? Don't need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job. To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

### Proposed Solution

S. No.	Parameter	Description
--------	-----------	-------------

1.	Problem Statement (Problem to be solved)	Job Seekers need an efficient job recommender system so that they get jobs based on their skills.
2.	Idea / Solution description	<p>We propose a skill/job recommender application that works as an intermediate between Job Seekers and Hiring Companies. The application lets the organizations register with job vacancies, along with the skills they expect for the candidates to possess. Similarly, the job seekers can register with their resume containing their skills. The solution involves,</p> <ol style="list-style-type: none"> <li>1. Hybrid filtering based on user skills.</li> <li>2. Filtering companies based on employees' reviews.</li> <li>3. Learning and quantifying recommendation abilities.</li> </ol>
3.	Novelty / Uniqueness	Suggesting / Providing study materials and courses in order to gain practical knowledge and on commonly expected skills in the market.
4.	Social Impact / Customer Satisfaction	The whole tiring process of searching for jobs suitable for one is totally eradicated and will be available at their fingertips through this app. This may decrease unemployment considerably. Getting notifications about new job vacancies and interviews increases customer convenience.
5.	Business Model (Revenue Model)	By providing companies with the best profiles, we can generate revenue by offering subscription-based applications to job seekers.
6.	Scalability of the Solution	This solution navigates through huge collections of items/jobs data and it extends up to the actual datasets, deals with efficiency and analysis to accelerate for massive datasets.

### 3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC	<b>1. CUSTOMER SEGMENT(S)</b> <span>CS</span> Who is your customer? i.e. working parents of 0-5 y.o. kids  1. Graduate Students 2. Working Professionals 3. Job Seekers	<b>6. CUSTOMER CONSTRAINTS</b> <span>CC</span> What resources prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connections, available devices.  1. Confidence 2. Premium Section 3. Spam Job alerts	<b>5. AVAILABLE SOLUTIONS</b> <span>AS</span> Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What price & more do these solutions have? i.e. pen and paper is an alternative to digital networking.  1. Daily Job Alert 2. Hiring Workflow 3. Segregation of Job based on Fields	Explore AS, differentiate
	<b>2. JOBS-TO-BE-DONE / PROBLEMS</b> <span>J&amp;P</span> Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one, explore different sides.  1. Premium Policy in the existing solution 2. Searching is to be made simple 3. Spam to be reduced	<b>9. PROBLEM ROOT CAUSE</b> <span>RC</span> What is the root reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.  There are various spam and fake job posting in the existing things the filter help the customers to easy navigate	<b>7. BEHAVIOUR</b> <span>BE</span> What does your customer do to address the problem and get the job done? (If directly related) find the right solar panel installer, calculate usage and benefits, indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)  Customer get their job done by accessing various platform and consulting firms.	
Identify strong TR & EM	<b>3. TRIGGERS</b> <span>TR</span> What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news. 1. Advertisement    2. More getting jobs 3. Easy access  <b>4. EMOTIONS: BEFORE / AFTER</b> <span>EM</span> How do customers feel when they face a problem or a job and afterwards? i.e. Not, insecure + confident, in control - use it in your communication strategy & design.  Before 1. No awareness about jobs 2. Applying for all jobs they get  After 1. Aware about the jobs 2. getting alerts about the jobs 3. more confident about getting a job	<b>10. YOUR SOLUTION</b> <span>SL</span> If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working in a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.  To give a end to end solution from applying a job to getting a job and give the API and lot of filters to get desired result and remove the spam jobs.	<b>8. CHANNELS of BEHAVIOUR</b> <span>CH</span> <b>8.1 ONLINE</b> What kind of actions do customers take online? Extract online channels from #7  <b>8.2 OFFLINE</b> What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.  Online 1. Search for job 2. Update the resume 3. apply for job  Offline 1. Visit the company 2. Go for interview	Identify strong TR & EM

## 4. REQUIREMENT ANALYSIS

### 4.1 FUNCTIONAL REQUIREMENT:



Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration (mobile users)	Registration can be done through Web application using mobile number or gmail
FR-2	User accessibility	The users needs to give access to location, storage and media and files.
FR-3	User profile	The users have to create a profile which has some basic information needed by the application
FR-4	User uploads	The users have to upload the softcopy of their mark sheet, identity card and resume.
FR-5	User verification	The user has to verify whether the given information are correct or not.
FR-6	End user benefits	This makes the recruiting process easy and also helps us to know the requirements in an effective way

## 4.2 NON-FUNCTIONAL REQUIREMENTS:

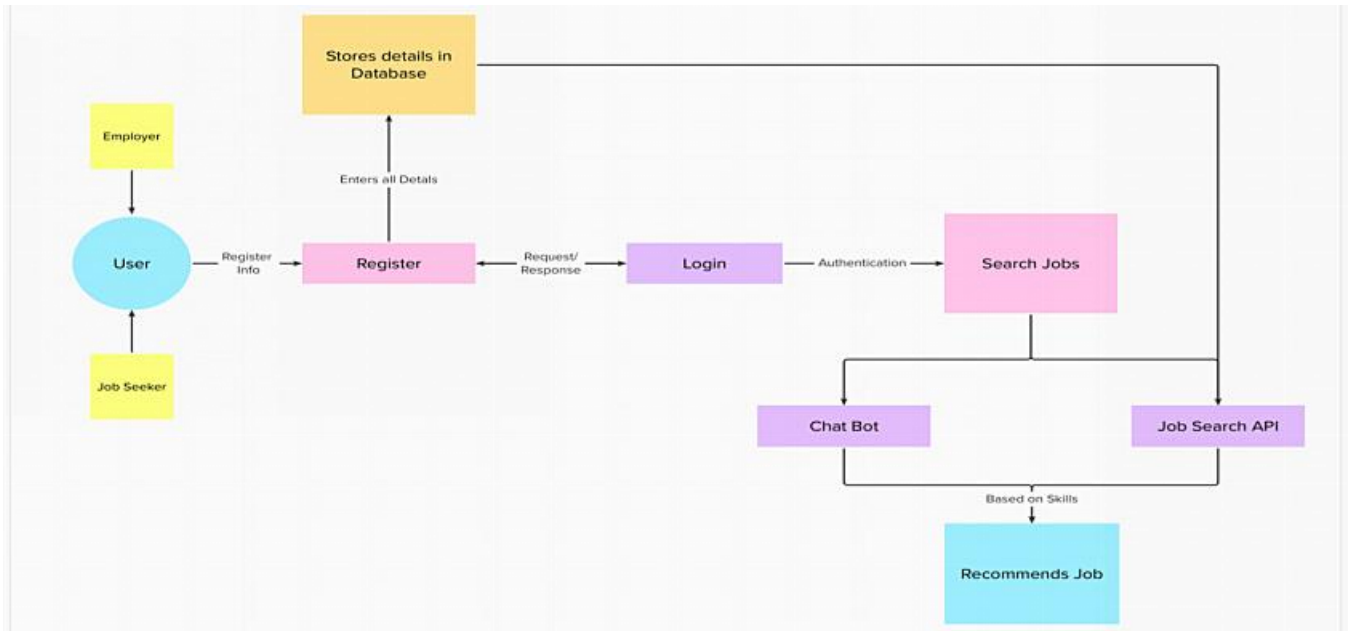
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	It is effective way to achieve the skill / job recommendation. It is easily access by everyone.
NFR-2	Security	The privacy of the users should be safe in the system.
NFR-3	Reliability	Integrity and consistency of the recommender engine and all its transactions should be ensured.
NFR-4	Performance	The recommender engine should generate recommendation within a time frame of 500 milliseconds.
NFR-5	Availability	It is always available in all platforms through website.
NFR-6	Scalability	It is convenient for the user to use the application and also this app has been considered user friendly.
NFR-7	learnability	A new user should be able to use the recommender engine with ease. In case of any queries, there must be some help provided to solve their queries.

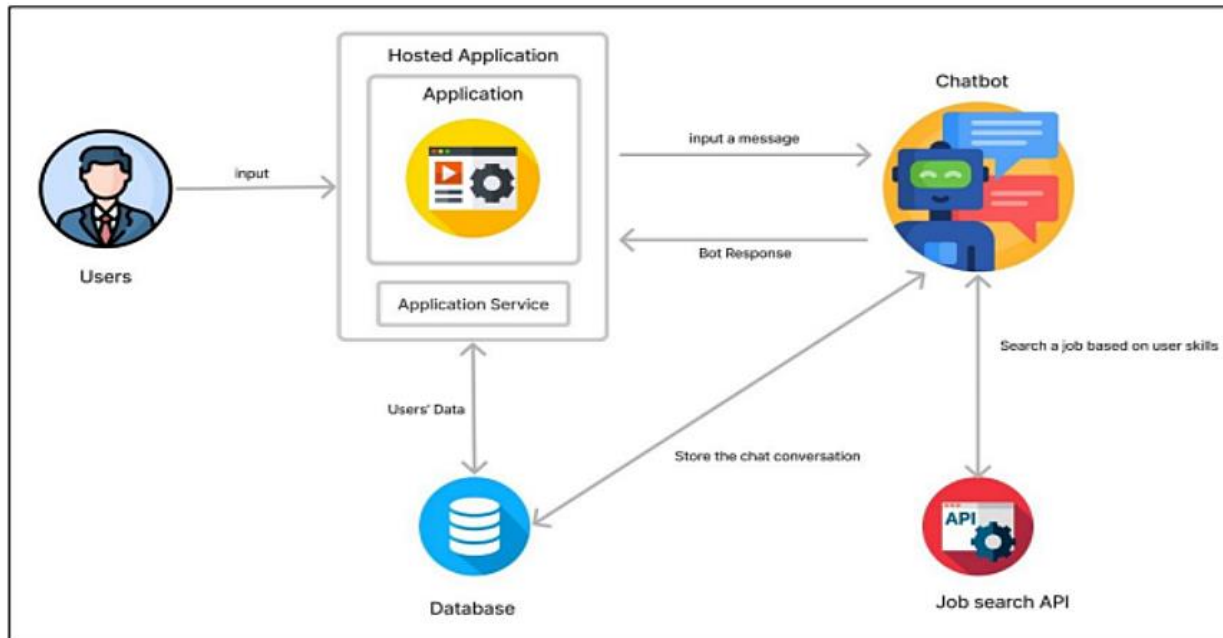
## 5.PROJECT DESIGN :

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 leaves the system, what changes the information, and where data is stored.

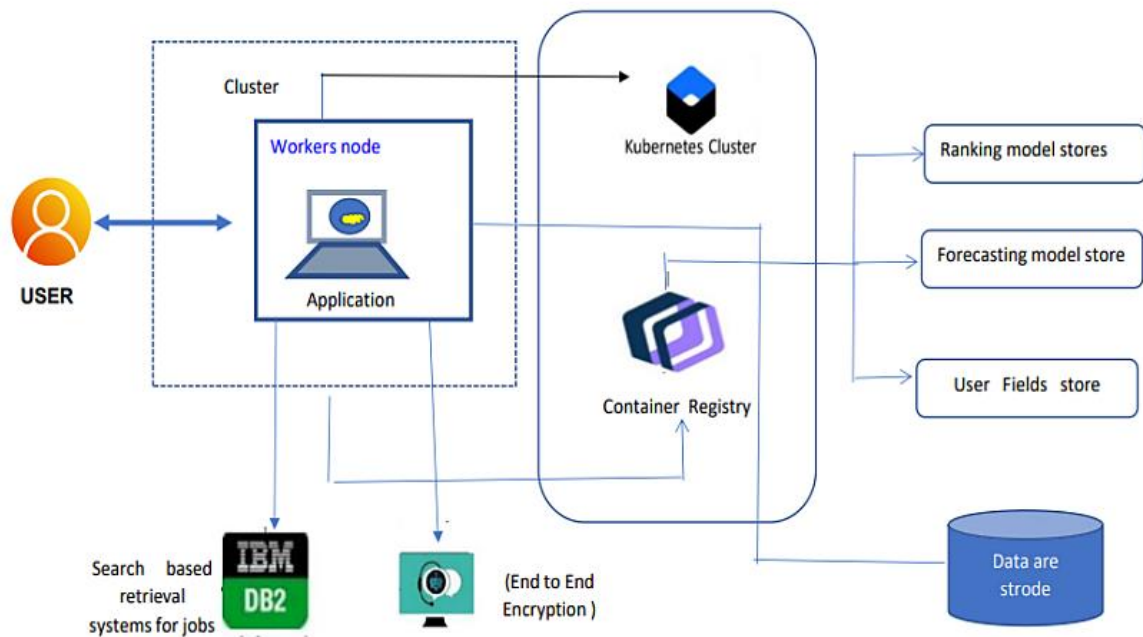


## 5.2 SOLUTION ARCHITECTURE:



## 5.3 TECHNICAL ARCHITECTURE:

### Technical Architecture:



**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 / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

## 5.4 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through LinkedIn	I can register & <u>access</u> the dashboard	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login and view the dashboard	High	Sprint-1
	Dashboard	USN-5	As a user, I can view and manage application	I can review company statistics	High	Sprint-1
Customer ( <u>mobile</u> user)	Registration & Login	USN-6	As a user, I can log into the application by entering email & password	I can login and view the dashboard	Low	Sprint-2

## 6. PROJECT PLANNING & SCHEDULING

## 6.1 Sprint Planning & Estimation

<b>Title</b>	<b>Description</b>	<b>Date</b>
<b>Literature Survey and Information Gathering</b>	Gathering Information by referring the technical papers, research publications etc	2 SEPTEMBER 2022
<b>Prepare Empathy Map</b>	To capture user pain and gains Prepare List of Problem Statement	10 SEPTEMBER 2022
<b>Ideation</b>	Prioritise a top 3 ideas based on feasibility and Importance	17 SEPTEMBER 2022
<b>Proposed Solution</b>	Solution include novelty, feasibility, business model, social impact and scalability of solution	24 SEPTEMBER 2022
<b>Problem Solution Fit</b>	Solution fit document	29 SEPTEMBER 2022
<b>Solution Architecture</b>	Solution Architecture	1 October 2022
<b>Customer Journey</b>	To Understand User Interactions and experiences with application	8 October 2022
<b>Functional Requirement</b>	Prepare functional Requirement	14 October 2022
<b>Data flow Diagrams</b>	Data flow diagram	15 October 2022

<b>Technology Architecture</b>	Technology Architecture diagram	16 October 2022
--------------------------------	---------------------------------	-----------------

<b>Milestone &amp; sprint delivery plan</b>	Activitywhatwedone &further plans	21 October 2022
<b>Project Development-Delivery of sprint 1,2,3&amp;4</b>	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

## 6.2 Sprint Delivery Schedule

Sprint	Functional Requirements	User story number	Description	Story Points	Priority	Team Members
Sprint 1	Login (Job Seeker)	USN-1	User must login to look for suitable job	20	High	4
Sprint 1	Login (Organisation)	USN-2	Oragnization can look for candidates matching their requirements	20	High	4
Sprint-2	Dashboard	USN-3	User can create their job profile and look for job attending eligible test. If user gets required marks, they will be invited to participate in interview	20	Low	4
Sprint-3	Dashboard	USN-4	Organization can ook for candidate based on their skill set	20	Medium	4
Sprint-4	ChatBot	USN-5	If user faces any difficulties, they can get	20	High	4

			help from chatbot			
--	--	--	-------------------	--	--	--

### Project Tracker, Velocity & Burn down Chart:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End date (Planned)	Story points completed	Sprint release date
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	05Nov 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

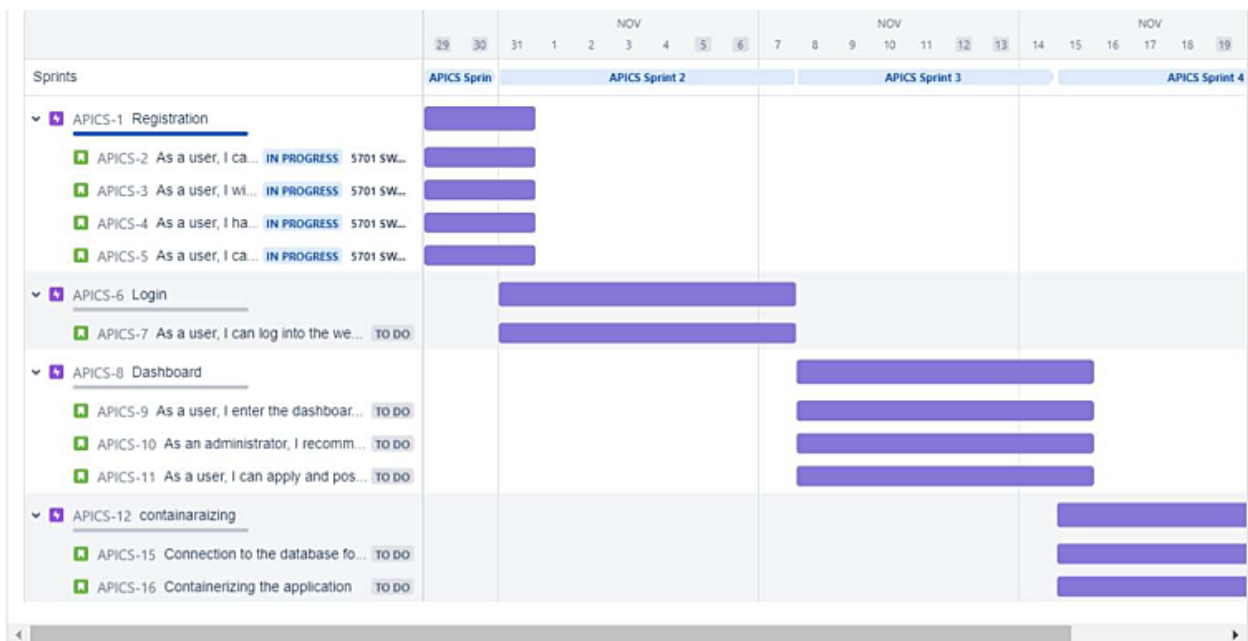
### VELOCITY:

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) per iteration unit (story points per day)

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



## 6.3 REPORTS FROM JIRA



## 7. CODING & SOLUTIONING

## 1. Self-service On-Demand

This is one of the most essential and significant characteristics of cloud computing. This means that cloud computing enables clients to regularly monitor the abilities, allotted network storage, and server uptime. Therefore, it is one of the most fundamental features of cloud computing that helps clients control various computing abilities as per their requirements.

## 2. Resources Pooling

This is also a fundamental characteristic of cloud computing. Pooling resources means that a cloud service provider can distribute resources for more than one client and provide them with different services according to their needs. Resource Pooling is a multi-client plan useful for data storing, bandwidth services and data processing services. The provider administers the data stored in real-time without conflicting with the need for data.

## 3. Easy Maintenance

This is one of the best cloud characteristics. Cloud servers are easy to maintain with low to almost zero downtime. Cloud Computing powered resources undergo several updates frequently to optimize their capabilities and potential. The updates are more viable with the devices and perform quicker than the previous versions.

## 4. Economical

This kind of service is economical as it efficiently reduces IT costs and data storage expenditure. Moreover, most cloud computing services are free. Even if there are paid plans, it's only to expand storage capacity, and these costs are often very nominal. This is a massive advantage of using cloud computing services.

## 5. Rapid Elasticity and Scalability

The best part of using cloud storage is that it can easily handle all the workload and data load concerning storage. Furthermore, as it is fully automated, businesses and organizations can save heavily on manual labor and technical staffing as cloud services are elastic, scalable and automated. This is one of the significant advantages of using cloud services.

## API Features:

1. HTTPS/SSL certificates Programming cheat sheets
2. Try for free: Red Hat Learning Subscription
3. eBook: An introduction to programming with Bash

Bash Shell Scripting Cheat Sheet

#### 4. eBook: Modernizing Enterprise Java

The gold standard for the web is HTTPS using SSL certificates, and Let's Encrypt can help you achieve this. It is a free, automated, and open certificate authority from the non-profit Internet Security Research Group (ISRG).

#### 2. Cross-origin resource sharing

CORS is a browser-specific security policy preflight check. If your API server is not in the same domain as the requesting client's domain, you will need to deal with CORS. For example, if your server is running on `api.domain-a.com` and gets a client request from `domain-b.com`, Coarsens an HTTP precheck request to see if your API service will accept client-side requests from the client's domain,

#### 3. Authentication and JSON Web Tokens

There are several approaches to validate an authenticated user in your API, but one of the best ways is to use JSON Web Tokens (JWT). These tokens are signed using various types of well-known cryptographic libraries. When a client logs in, an identity-management service provides the client with a JWT. The client can then use this token to make requests to the API. The API has access to a public key or a secret that it uses to verify the token. There are several libraries available to help verify tokens, including `honeytoken`. For more information about JWT and the libraries that support it in every language, check out `JWT.io`.

#### 4. Authorizations and scopes

Authentication (or identity verification) is important, but so is authorization, i.e., does the verified client have the privilege to execute this request? This is where scopes are valuable. When the client authenticates with the identity management server and a JWT token is created, having the identity management service provide the scopes for the given authenticated client can enable the API service to determine if this verified client request can be performed without having to perform an additional costly lookup to an access control list.

### 7.2 FEATURE 2

#### Docker Features:

##### 1. Faster and Easier configuration:

It is one of the key features of Docker that helps you in configuring the system in a faster and easier manner. Due to this feature, codes can be deployed in less time and with fewer efforts. The infrastructure is not linked with the environment of the application as Docker is used with a wide variety of environments.

##### 2. Application isolation:

Docker provides containers that are used to run applications in an isolated environment. Since each container is independent, Docker can execute any kind of application.

### 3. Increase in productivity:

It helps in increasing productivity by easing up the technical configuration and rapidly deploying applications. Moreover, it not only provides an isolated environment to execute applications, but it reduces the resources as well.

### 4. Swarm:

Swarm is a clustering and scheduling tool for Docker containers. At the front end, it uses the Docker API, which helps us to use various tools to control it. It is a self-organizing group of engines that enables pluggable backends.

### 5. Services:

Services is a list of tasks that specifies the state of a container inside a cluster. Each task in the Services lists one instance of a container that should be running, while Swarm schedules them across the nodes.

## Kubernetes Features:

1. Auto-scaling. Automatically scale containerized applications and their resources up or down based on usage

2. Lifecycle management. Automate deployments and updates with the ability to:

- a. Rollback to previous versions

- b. Pause and continue a deployment

3. Declarative model. Declare the desired state, and K8s works in the background to maintain that state and recover from any failures.

4. Resilience and self-healing. Auto placement, auto restart, auto replication and auto scaling provide application self-healing

5. Persistent storage. Ability to mount and add storage dynamically

6. Load balancing. Kubernetes supports a variety of internal and external load balancing options to address diverse needs.

.

## 7.3 DATABASE SCHEMA

Username: Varchar (32)

Email: Varchar (32)

Phone Number: Varchar (32)

Password: Varchar (32)

Pin: Varchar

## 8. TESTING

1. Verify that after registration users are navigated to login page
2. Verify the UI elements in Login/Signup popup
3. Verify user can log into application with Valid credentials
4. Verify that categories of news are shown in homepage
5. Verify that news is displayed in homepage
6. Verify that when clicked on news it is redirected to correct page

### 8.2 USER ACCEPTANCE TESTING

The purpose of this documents to briefly explain the test coverage and open issues of the Recommender Application project at the time of the release User Acceptance Testing (UAT). This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

#### 1.Defect Analysis

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

#### 2.Test Case Analysis

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

## 9. RESULT

### 9.1 PERFORMANCE METRICS

The application performance index, or Apex 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.

Those transactions are then bucketed into satisfied (fast), tolerating (sluggish), too slow, and failed requests. A simple math formula is then applied to provide a score from 0 to 1.



## Registration

Email ID

Username

Password

Enter Phone number:

Enter four digit pin:

Registration successfull. Log in to continue |

submit

[Back to page](#)

indeed.com

## Login

Username

nthsh

Password

\*\*\*

submit

[Forgot your password? Try Another Way](#)

[Don't have an account? Create new account](#)

indeed.com

## APPLY JOBS



APPLY



APPLY



APPLY





### Our Courses



#### [HTML5 For Beginners](#)

This course was designed for students starting out in Front End Web Development wanting to learn HTML5 to get started.....



#### [CSS3 For Beginners](#)

This course was designed for students starting out in Front End Web Development wanting to learn CSS3 to get started.....



#### [JavaScript For Beginners](#)

This course was designed for students starting out in Front End Web Development wanting to learn JavaScript to get started.....

### ABOUT US

Hello Everyone! We did this project for IBM Nalalyathiran. Our team consists of four members. Our Project is: Skill/Job Recommender Application. This project recommends Jobs for Job seekers and also skills. We heartfully Thank IBM for providing their free resources and we gained lot of knowledge from learning new technologies.

#### TEAM :

TEAM LEADER : Roshan Adhithya SS

TEAM MEMBER 1 : Vignesh M

TEAM MEMBER 2 : Nithish Bharathwaj

TEAM MEMBER 3 : Rishikesh B

[Home](#) [Register](#) [Login](#)

indeed.com

Contact us

first name

last name

email

text area

submit

[Home](#) [Register](#) [Login](#) [Apply Job](#) [Skills](#) [About us](#) [contact us](#)

indeed.com

Search Jobs

Search



Top 50 Best Job Search Sites

SKYS JOB PORTAL

Choose company preference

service product

service

Here some of the service-based company

wipro tcs cts Accenture

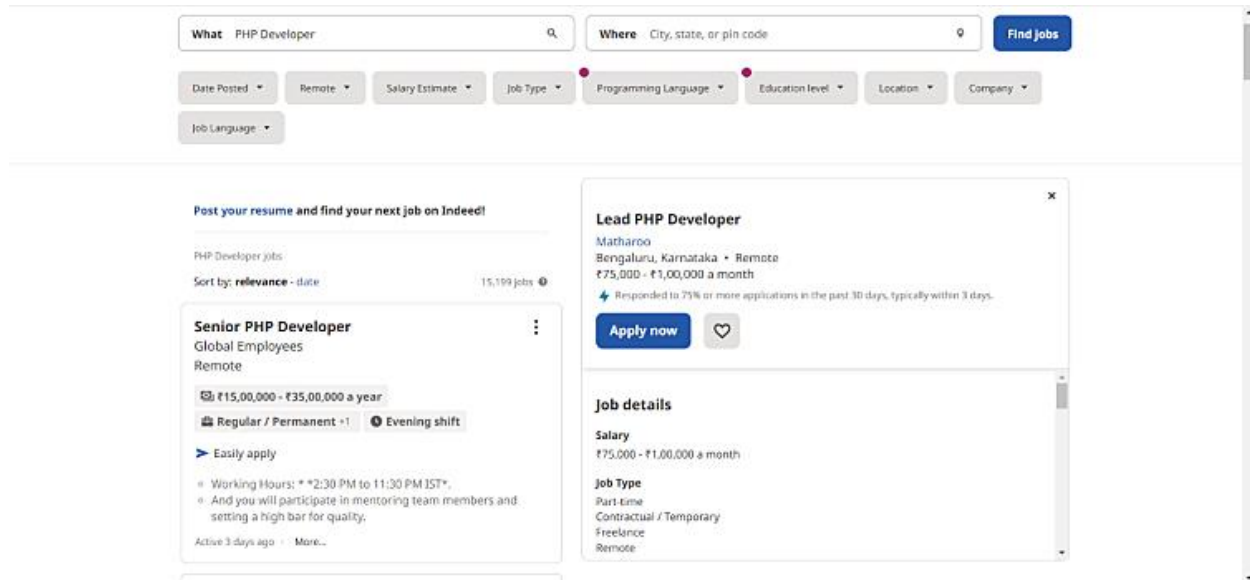
wipro

choose a job role

Select an option

Type something...

Built with IBM Watson®



## 10.ADVANTAGES AND DISADVANTAGES

### 1.Employment Opportunities:

The foremost advantage of having a profile in our application is that it is your doorway into employment opportunities worldwide. Before the advent of online job applications, students would get jobs through connections. However, now your job opportunities have increased magnanimously. Students who have attained education abroad can put in their area of specialization and find an appropriate job. Apart from this, if there is a particular company that you're interested in, you can make applications for the same.

### 2.Easy Job Applications:

The traditional recruiting process has taken a back seat and online job application has become paramount. Gone are the days, where you would have to run around with copies of your resume. With the ease of uploading the necessary information on your profile, not only will the recruiters peruse through your profile, but you can update your skills regularly. The initial stress of a job application is reduced because the recruiter is already aware of your skills and wants to explore them further. This gives you an excellent opportunity to capitalize on the same and use the app to its fullest.

### 3. Initiate Connections:

Apart from receiving a job offer, the connections you establish on your profile help you in the long run. For instance, you may start by connecting with your school and college friends and eventually shift to your colleagues. An alumnus from your university is good connections to have. Having an illustrious list of connections speaks to your strong profile. Having a connection who is working at your dream company can be your pathway to the same. Initiating connections will allow you to analyze industry trends and be at the top of the game.

### 4. Endorsement and Connections

Collecting endorsements and connections is an excellent way of adding social backing to your profile. As mentioned earlier, having illustrious connections will add value to your profile. Upon receiving endorsement for your skills, employers receive extra confidence in your profile. The trick now is to not only have relevant skills but also make your profile stand out.

#### Disadvantages:

#### 1. Risk of identity theft

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

#### 2. Incomplete profile challenge

LinkedIn like other social network websites required 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.

#### 3. Tons of spam messages

There's a saying that among 12 disciples there will always be a Judas. Think of how many Judas will be available on a website with over 1200 million people. LinkedIn is filled with spam messages from recruiters, employers, and even job seekers. All just to seek attention, mislead, and extort money, etc.

#### 4. Premium package can be expensive

Good thing they say doesn't come cheap. Although, LinkedIn allows you to join the platform without paying. But the LinkedIn premium packages are charged for. For example, the "medium-sized career" price is just about \$29.99/month. There are so many added benefits that this offer brings but can still be very costly for a starter or medium-sized business.

## 11.CONCLUSION

We proposed an application for job recommendation task. This application facilitates the opportunities of job recommendation process as well as it allows the use of a variety of text processing and recommendation methods according to the preferences of the job recommender system designer. Moreover, we also contribute making publicly available a new dataset containing job seekers profiles and job vacancies. 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.

## 12. FUTURE SCOPE

The future is and will remain unknown to us , but fact is there is a issue of unemployment in the world of job market, the job portals will exist and will grow in proportionate with demand

## 13.APPENDIX

Source Code:

### 1.Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>indeed.com</title>
  <link rel="stylesheet" href="{url_for('static', filename='style.css')}}">
  <script>
    window.watsonAssistantChatOptions = {
      integrationID: "5bea080a-0793-42c5-97b8-3ede9c7b6a0c", // The ID of this integration.
      region: "us-south", // The region your integration is hosted in.
      serviceInstanceID: "ead2bc3a-7163-45a7-ae03-bd16c50d4ecc", // 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>
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{url_for('home')}}">Home</a></li>
                <li><a href="{{url_for('register')}}">Register</a></li>
                <li><a href="{{url_for('login')}}">Login</a></li>
                <li><a href="{{url_for('welcome_page')}}">Apply Job</a></li>
                <li><a href="{{url_for('skills')}}">Skills</a></li>
                <li><a href="{{url_for('about')}}">About us</a></li>
                <li><a href="{{url_for('contact')}}">contact us</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <h1 class="two" ><span style="color:rgb(20, 20, 70);background-color: whitesmoke;margin-left:
30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1>
        <div id="ss1">
            <input class="input_search" type="text" placeholder="Search Jobs" >
            <button class="button_search">Search</button>
        </div>
        <center></center>
    </section>

</body>
</html>

```

## 2.Login.html:

```
<html>
```

```

<head>
  <title>Login page</title>
  <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Back to page</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1></center>
  </section>
  <br><br>
  <div class="container"> <br /><br />
  <form action="/login" method="POST">
    <h1>Login</h1> <br /><br />
    <label class="form_label"for="email"><b>Username</b></label><br><br>
    <input class="form_input"type="text" name= "username" /><br><br>
    <label class="form_label"for="psw"><b>Password</b></label><br><br>
    <input class="form_input"type="password" name="password"/>
    <br><br><br>
    <center><input type="submit" class="submitbtn"value="submit" /></center>
  </form>
</div>
  <center><p>Forgot your password? <a href="{{url_for('forget')}}">Try Another Way</a></p></center>
  <center><p>Don't have a account <a href="{{url_for('register')}}">Create new
account</a></p></center>
</body>
</html>

```

### 3.Register.html

```

<html>
  <head>
    <title>Registration page</title>

```



```

<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{ url_for('home') }}">Back to page</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:white;margin-left:
30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1></center>
    </section>
    <div class="container">
        <form action="/register" method="POST">
            <h2>Registration</h2>
            <label class="form_label" for="email"><b>Email ID</b></label><br><br>
            <input class="form_input" type="email" name="email"/><br><br>
            <label class="form_label" for="user"><b>Username</b></label><br><br>
            <input class="form_input" type="text" name="username" /><br><br>
            <label class="form_label" for="psw"><b>Password</b></label><br><br>
            <input class="form_input" type="password" name="password"/><br><br>
            <label class="form_label" for="pho"><b> Enter Phone number:</b></label><br><br>
            <input class="form_input" type="text" name="phonenumber"/><br><br>
            <label class="form_label" for="pho"><b> Enter four digit pin:</b></label><br><br>
            <input class="form_input" type="password" name="pin"/>
            <br><br><br>
            <center> {% if error %}
            <p><strong style="color:red">Error</strong>: {{error}}</p>
            {% endif %}
            {% with messages = get_flashed_messages() %}
            {% if messages %}
                {% for message in messages %}
                    <p style="color:green">{{ message }}</p>
                {% endfor %}
            {% endif %}
            {% endwith %} </center>
            <center><input type="submit" class="submitbtn" value="submit" /></center>
            <center><p>Already have a account <a href="{{ url_for('login') }}">Sign in</a></p></center>
        </form>

```

```
</div>
</body>
</html>
```

## 4.skills.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SKYS.com</title>
  <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Home</a></li>
        <li><a href="{{url_for('register')}}">Register</a></li>
        <li><a href="{{url_for('login')}}">Login</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1></center>
  </section>
  <br><br>
  <section id="courses">

    <center><h1>Our Courses</h1></center>

    <div class="course">
      
```

```
<center><h2><a href="https://www.w3schools.com/html/">HTML5 For  
Beginners</a></h2></center>
```

```
<center><h6>This course was designed for students starting out in Front End Web Development  
wanting to learn HTML5 to get started.....</h6></center>
```

```
</div>
```

```
<div class="course2">
```

```

```

```
<center><h2><a href="https://www.w3schools.com/css/">CSS3 For Beginners</a></h2></center>
```

```
<center><h6>This course was designed for students starting out in Front End Web Development  
wanting to learn CSS3 to get started.....</h6></center>
```

```
</div>
```

```
<div class="course3">
```

```

```

```
<center><h2><a href="https://www.w3schools.com/js/">JavaScript For Beginners</a></h2></center>
```

```
<center><h6>This course was designed for students starting out in Front End Web Development  
wanting to learn JavaScript to get started.....</h6></center>
```

```
</div>
```

```
</section>
```

```
</body>
```

```
</html>
```

## 5.welcome.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SKYS.com</title>
  <link rel="stylesheet" href="{ {url_for('static', filename='style.css')}}">
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{ {url_for('home')}}">Home</a></li>
        <li><a href="{ {url_for('register')}}">Register</a></li>
        <li><a href="{ {url_for('login')}}">Login</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1></center>
  </section>
  <br><br>
  <section id="jobs">

    <center><h1><b>APPLY JOBS</b></h1></center><br>
    
    
    
    
```

```



<h2 style="position:absolute;top: 450px;left:150px"><a href="https://in.indeed.com/PHP-Developer-
jobs?vjk=10cca9575b193c7d"><button>APPLY</button></a></h2>
<h2 style="position:absolute;top: 450px;left:650px"><a href="https://in.indeed.com/Software-Developer-
jobs?vjk=b7da08f07cac87d5"><button>APPLY</button></a></h2>
<h2 style="position:absolute;top: 450px;left:1150px"><a href="https://in.indeed.com/Web-Developer-
jobs?vjk=b81e49165da51eeb"><button>APPLY</button></a></h2>
<h2 style="position:absolute;top: 750px;left:1150px"><a href="https://in.indeed.com/SQL-Developer-
jobs?vjk=ac86b15908022123"><button>APPLY</button></a></h2>
<h2 style="position:absolute;top: 750px;left:650px"><a href="https://in.indeed.com/Java-Developer-
jobs?vjk=da306a665e00eb30"><button>APPLY</button></a></h2>
<h2 style="position:absolute;top: 750px;left:150px"><a href="https://in.indeed.com/Python-Developer-
jobs?vjk=fa7b9bd250044569"><button>APPLY</button></a></h2>
</section>
</body>
</html>

```

## 6.Forgot.html

```

<html>
<head>
<title>Login page</title>
<link rel="stylesheet" href="{ {url_for('static', filename='style.css') }}">
</head>
<body>
<header>
<nav class="navbar">
<ul>
<li><a href="{ {url_for('welcome_page') }}">Back to page</a></li>
</ul>
</nav>
</header>
<section id="sss1">
<center><h1 class="two" ><span style="color:rgb(109, 30, 192);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">indeed</span>.com</h1></center>
</section>
<br><br>

```

```

{% if error %}
    <p><strong style="color:red">Error</strong>: {{ error }}</p>
{% endif %}
{% with messages = get_flashed_messages() %}
    {% if messages %}
        {% for message in messages %}
            <p style="color:green">{{ message }}</p>
        {% endfor %}
    {% endif %}
{% endwith %}
<div class="container"> <br /><br />
<form action="/forget" method="POST">
<h1>Try to login with your 4 digit pin</h1> <br /><br />
    <label class="form_label" for="email"><b>Username</b></label><br><br>
    <input class="form_input" type="text" name="username" /><br><br>
    <label class="form_label" for="psw"><b>Pin</b></label><br><br>
    <input class="form_input" type="password" name="pin" />
    <br><br><br>
    <center><input type="submit" class="submitbtn" value="submit" /></center>
</form>
</div>
<center><p>Back to <a href="{{ url_for('login')}}">login</a></p></center>
</body>
</html>

```

## 7.Contact.html

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>indeed.com</title>
    <link rel="stylesheet" href="{{ url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">

```

```
<ul>
  <li><a href="{{url_for('home')}}">Home</a></li>
  <li><a href="{{url_for('register')}}">Register</a></li>
  <li><a href="{{url_for('login')}}">Login</a></li>
</ul>
</nav>
</header>
<section id="sss1">
  <center><h1 class="two" ><span style="color:rgb(20, 20, 70) ;background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1></center>
</section>
<br><br>
<center><h1 class="h1lt">Contact us</h1></center>

<form>

  <label class="form_label"><b>first name</b></label><br><br>
  <input class="form_input" type="text" name="firstname" value="">
<br><br>

<label class="form_label"><b>last name</b></label><br><br>
<input class="form_input" type="text" name="lastname" value="">

<br><br>

<label class="form_label"><b>email</b></label><br><br>
<input class="form_input" type="text" name="email" value="">
<br>
<br>

<label class="form_label"><b>text area</b></label><br><br>
<textarea class="form_input" type="text" name="text area" value=""></textarea>
<br><br>
<center><input type="submit" class="submitbtn" value="submit" /></center>
</form>

</body>
</html>
```



## 8.About.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>indeed.com</title>
  <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Home</a></li>
        <li><a href="{{url_for('register')}}">Register</a></li>
        <li><a href="{{url_for('login')}}">Login</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <center><h1 class="two" ><span style="color:rgb(20, 20, 70) ;background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">indeed.com</span></h1></center>
  </section>
  <br><br>
  <center><h1 class="h1lt">Contact us</h1></center>

  <form>

    <label class="form_label"><b>first name</b></label><br><br>
    <input class="form_input" type="text" name="firstname" value="">
  <br><br>

  <label class="form_label"><b>last name</b></label><br><br>
  <input class="form_input" type="text" name="lastname" value="">

  <br><br>

  <label class="form_label"><b>email</b></label><br><br>
  <input class="form_input" type="text" name="email" value="">
```

```

<br>
<br>

<label class="form_label"><b>text area</b></label><br><br>
<textarea class="form_input" type="text" name="text area" value=""></textarea>
<br><br>
<center><input type="submit" class="submitbtn" value="submit" /></center>
</form>

</body>
</html>

```

## App.py

```

from flask import Flask, render_template, request, redirect, url_for, session, flash
import ibm_db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
import requests

app=Flask(__name__, template_folder='Templates')
app.secret_key='a'
try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SSL;SSLServ
erCertificat=DigiCertGlobalRootCA.crt;UID=ynd42279;PWD=gygul6zNcomD2Bxj", "", "")
except:
    print("Unable to connect: ", ibm_db.conn_error())

@app.route("/")
def dash():
    return render_template('home.html', msg=" ")

@app.route("/register", methods=['GET', 'POST'])
def register():
    error = None
    if request.method == 'POST':
        username=request.form['username']
        email=request.form['email']

```

```

phone_number=request.form['phonenumber']
password=request.form['password']
pin=request.form['pin']
sql="SELECT * FROM user WHERE phone_number=?"
prep_stmt=ibm_db.prepare(conn,sql)
ibm_db.bind_param(prepare_stmt,1,phone_number)
ibm_db.execute(prepare_stmt)
account=ibm_db.fetch_assoc(prepare_stmt)
print(account)

#message = Mail(from_email='btechmano@gmail.com',to_emails=session['email'],subject="Devnews
- Registration",html_content='<b>Devnews welcomes you</b><br/><p>Your account has been registered
successfully</p>')

#try:
#sg=SendGridAPIClient()
# Secret key can't be submitted otherwise my
# sendgrid account reporting that i am exposing
# my secret key as public and my account will terminated soon
#response=sg.send(message)
#print(response.status_code)
#print(response.body)
#print(response.headers)
#except Exception as e:
#print(e)

if account:
    error="Account already exists! Log in to continue !"
else:
    insert_sql="INSERT INTO user values(?,?,?,?)"
    prep_stmt=ibm_db.prepare(conn,insert_sql)
    ibm_db.bind_param(prepare_stmt,1,email)
    ibm_db.bind_param(prepare_stmt,2,username)
    ibm_db.bind_param(prepare_stmt,3,phone_number)
    ibm_db.bind_param(prepare_stmt,4,password)
    ibm_db.bind_param(prepare_stmt,5,pin)
    ibm_db.execute(prepare_stmt)
    flash(" Registration successfull. Log in to continue !")

else:
    pass
return render_template('register.html',error=error)

@app.route('/login',methods=['GET','POST'])
def login():

```

```

error = None
if request.method=='POST':
    username=request.form['username']
    password=request.form['password']
    sql="SELECT * FROM user WHERE username=? AND password=?"
    stmt=ibm_db.prepare(conn,sql)
    ibm_db.bind_param(stmt,1,username)
    ibm_db.bind_param(stmt,2,password)
    ibm_db.execute(stmt)
    account=ibm_db.fetch_assoc(stmt)
    print(account)
    if account:
        session['Loggedin']=True
        session['id']=account['USERNAME']
        session["username"]=account["USERNAME"]
        flash("Logged in successfully!")
        return redirect(url_for("home"))
    else:
        error="Incorrect username / password"
        return render_template('login.html',error=error)
return render_template('login.html',error=error)

```

```

@app.route('/forget',methods=['GET','POST'])

```

```

def forget():

```

```

    error = None
    if request.method=='POST':
        username=request.form['username']
        pin=request.form['pin']
        sql="SELECT * FROM user WHERE username=? AND pin=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,pin)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session["username"]=account["USERNAME"]
            flash("Logged in successfully!")
            return redirect(url_for("home"))

```

```

    else:
        error="Incorrect username / pin"
        return render_template('login.html',error=error)
    return render_template('forget.html',error=error)
@app.route('/welcome')
def welcome_page():
    return render_template("welcome.html",msg=" ")
@app.route('/home')
def home():
    return render_template("home.html",msg=" ")
@app.route('/skills')
def skills():
    return render_template("skills.html",msg=" ")
@app.route('/about')
def about():
    return render_template("about.html",msg=" ")
@app.route('/contact')
def contact():
    return render_template("contact.html",msg=" ")
if __name__=='__main__':
    app.run(debug=True,host="0.0.0.0")

```

## Docker File:

```

FROM python:3.6
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python3 -m pip install -r requirements.txt
RUN python3 -m pip install ibm_db
EXPOSE 5000
CMD ["python","app.py"]

```

## Requirements File:

```

flask
ibm_db
sendgrid
requests

```

## Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: flask-node-deployment
spec:
  ports:
    - port: 5000
      targetPort: 5000
  selector:
    app: flasknode
```

## Deploy.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-node-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
        - name: flasknode
          image: us.icr.io/job-skill/job-skill
          imagePullPolicy: Always
          ports:
            - containerPort: 5000
```

**GitHub & Project Demo Link:**

**Github Link:**

<https://github.com/IBM-EPBL/IBM-Project-16891-1659624517>

**Demo Link:**

[https://drive.google.com/file/d/1bykjW0F0q59xJhwY-Tw9NXp0FV5vICvu/view?usp=share\\_link](https://drive.google.com/file/d/1bykjW0F0q59xJhwY-Tw9NXp0FV5vICvu/view?usp=share_link)