TEAM ID :PNT2022TMID46377

IBM-44252-1662637055

SKILL AND JOB RECOMMENDER

TEAM DETAILS:

Team ID **:** PNT2022TMID46377

College Name **:** A.V.C. COLLEGE OF ENGINEERING

Department **:** COMPUTER SCIENCE & ENGINEERING

TEAM MEMBERS:

1. **PASUPATHY G**
2. **GOPINATH B**
3. **ARUL M**
4. **RAMAPRABHAKARAN R**

TEAM ID :PNT2022TMID46377

TABLE CONTENTS

ABSTRACT

One domain where such recommender systems can play a significant role to help college graduates to fulfill their dreams by recommending a job based on their skill set. Currently, there are plenty of websites that provide heaps of information regarding employment opportunities, but this task is extremely tedious for students as they need to go through large amounts of information to find the ideal job. And many students are not aware of which job is suitable for them. Nowadays, the IT fields are in a boom. Many engineering students are learning some technical skills by doing some courses but they don't know which skill is for which job. Simultaneously, existing job recommendation systems only take into consideration the domain in which the user is interested while ignoring their profile and skillset, which can help recommendobs that are tailor-made for the user. This paper examines the user's resume then compares the knowledge of degree, soft skills, hard skills, and the projects he has done and then only the system recommends the jobs for that user. The system not only recommends the jobs but also shows the score of his/her resume for the respective job. Then, the system also recommends skills to improve the scores of theirMachine learning is a subfield of data science that concentrates on designing algorithms that can learn from and make predictions on the data. Presently recommendation frameworks are utilized to take care of the issue of the overwhelming amount of information in every domain and enable the clients to concentrate on information that is significant to their area of interest. One domain where such recommender systems can play a significant role to help college graduates to fulfill their dreams by recommending a job based on their skill set. Currently, there are plenty of websites that provide heaps of information regarding employment opportunities, but this task is extremely tedious for students as they need to go through large amounts of information to find the ideal job. And many students are not aware of which job is suitable for them. Nowadays, the IT fields are in a boom. Many engineering students are learning some technical skills by doing some courses but they don't know which skill is for which job. Simultaneously, existing job recommendation systems only take into consideration the domain in which the user is interested while ignoring their profile and skillset, which can help recommend jobs that are tailor-made for the user. This paper examines the user's resume then compares the knowledge of degree, soft skills, hard skills, and the projects he has done and then only the system recommends the jobs for that user. The system not only recommends the jobs but also shows the score of his/her resume for the respective job. Then, the system also recommends skills to improve the scores of theirMachine learning is a sub-field of data science that concentrates on designing algorithms that can learn from and make predictions on the data. Presently recommendation frameworks are utilized to take care of the issue of the overwhelming amount of information in every domain and enable the clients to concentrate on information that is significant to their area of interest. .

1.INTRODUCTION

INTRODUCTION TO PROJECT*:*

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 login and find the jobs by using 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 skillset of the users.The users and their information are stored in the Database.An alert is sent when there is an opening based on the user skillset.User will interact with the chatbot and can get the recommendations based on his skills.We can use job search API to get the current job openings in the market which will fetch the data directly from the webpage.

1.1 PROJECT OVERVIEW

To find suitable jobs and their scores, this application receives the resume and has a dataset for a job with their description. It will pre-process the resume and job description with the stop words and porter’s steamer. Then it reduces into a meaningfulbag of words. Now the application uses a of-id f reflectorized to convert a raw text into a matrix which makes it easy while compare. The main step is comparing the two bag words. For that, it uses the Cosine Similarity function, which is an angle dependentcalculation. By using cosine, it has a list of jobs in descending order with respect toscores.

The system will move on to the next progress which is finding the skills to be improved by the candidates. The system will take the resume and the skills dataset then compares both and display the skills which are all not in the resume. The major contribution of this work is as follows: The large MNC businesses use the mechanism currently in place for employment recommendations. The method is employed by businesses, not by regular people. If not, they will charge a small subscription fee to check the user's career options. The system functions for the average guy from city to village to modify this predicament. Because the students would look for

employment based on their own skills, this approach will reduce unemployment. This company will also grow more quickly, which will result in more job openings.

1.1 PURPOSE OF THE PROJECT:

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.

TEAM ID :PNT2022TMID46377

2.LITERATURE SURVEY

AUTHOR	PAPER TITLE	YEAR	JOURNAL	CRITICS
Ravita Mishra , Sheetal Rathi	Enhanced(DSSM) Technique for job recommendation	2021	King Saud University	The runtime of the system and the run time of moderate as compare to others with contextual feature
Sven Laumer, Fabian Gubler, Christian Maier and Tim Weitzel	Job Seekers' Acceptance of Job Recommender Systems: Results of an Empirical Study	2018	Hawaii International Conference on System Sciences	• The importance of trust has been highlighted future research studies can focus on the antecedents of trust • Users require lower level of trust
Shaha T. Al-Otaibi and Mourad Ykhlef	A survey of job recommender systems	2012	International Journal of the Physical Sciences	• Binary representation only. • Less attributes used. • No perfect measures
Shivraj Hulbatte1, Amit Wabale, Suraj Patil& Nikhilkumar Sathe	Enhanced Job Recommendation System	2018	ISSN(International Standard Serial Number)	• Key words search method. • One way recommendation.
Cornee de Ruijt and Sandjai Bhulai	Job Recommender Systems: A Review	2021	Vrije Universiteit Amsterdam	• To limit the scope of this literature review • Currently classified as MM-SE, are quite similar to cascade hybrids

Roshan G. Belsare and V. M. Deshmukh	Employment Recommendation System using Matching, Collaborative Filtering and Content Based Recommendation	2018	International Journal of Computer Applications Technology and Research Volume	• It only tries to match jobs based on certain parameters • Does not contain any personalized information
Deepali V Musale , Mamta K Nagpure , Kaumudini S Patil and Rukhsar F Sayyed4	Job Recommendation System Using Profile Matching And Web-Crawling	2016	International journal of Advance Scientfic Reearch AND Engineering Trends	• No relational aspects are included • Tools and technologies skills excluded.
Mohamme d korayem,lay ala pourjaf and Walid shalaby	Help me a find a job: A graph-based approach for job recommendation atscale	2017	IEEE International conference on Big data	• Less attributes used. • No perfect measures
Sidnooma Christian Kabore	Design and implementation of a recommender system as a module for Liferay portal	2012	Barcelona School of Computing (FIB), Univesity Polytechnic of Catalunya(UP C) Master in Information Technologies	we have faced during the evaluation phase is adapting the data to the context of the Global world labor corp.
Vinay Desai, Dheeraj Bahl, Shreekumar Vibhandik and Isra Fatma	Implementation of an Automated Job Recommendation System Based on Candidate Profiles	2017	International Research Journal of Engineering and Technology (IRJET)	The increased scores suggest that the improved recommender works well for the reason that jobs take precedence (1st and 2nd recommended jobs) are better than latter ones (3rd recommended job).

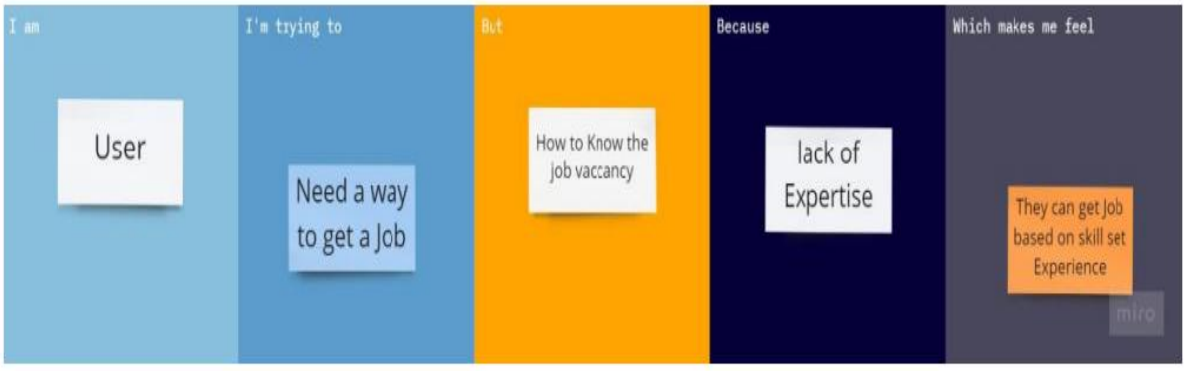
TEAM ID :PNT2022TMID46377

2.1 EXISTING SYSTEM:

In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personal-ized job suggestions. Although in the literature exists a variety of techniques and strategies used as part of job recommender systems, most of them fail to recommending job vacancies that fit properly to the job seekers profiles. Thus, the contributions of this work are threefold, we: i) made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites; ii) put forward the proposal of a framework for job recommendation based on professional skills of job seekers; and iii) carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. We thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue.

2.2 PROBLEM STATEMENT DEFINITION:

The problem statement aims at connecting the employer and job seekers where employers are the resource of the resources and the job seeker can find and apply for their targeted job. It allows registered users to be able to search for jobs and filter the results based on required skills, salary, experience level, etc.,



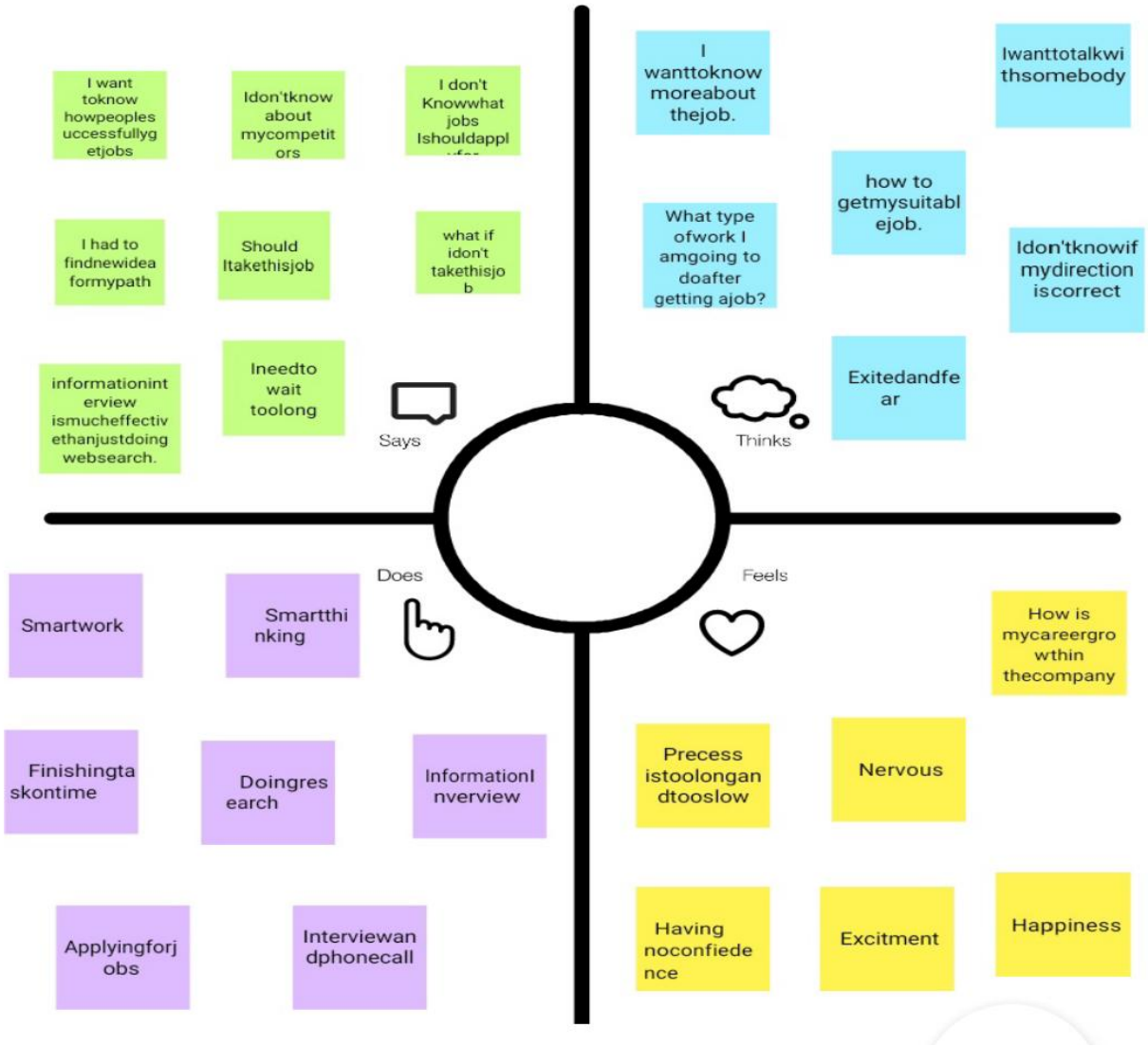
PROBLEM STATEMENT(PS)	***I am (Customer)***	***I'm trying to***	***But***	***Because***	***Which makes me feel***
PS-1	User	Need a way to get a job	How to know the job vacancy	Lack of Expertise	They can get job based on Skill set Experience

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

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

TEAM ID :PNT2022TMID46377



3.2 Ideation and Brainstorming:

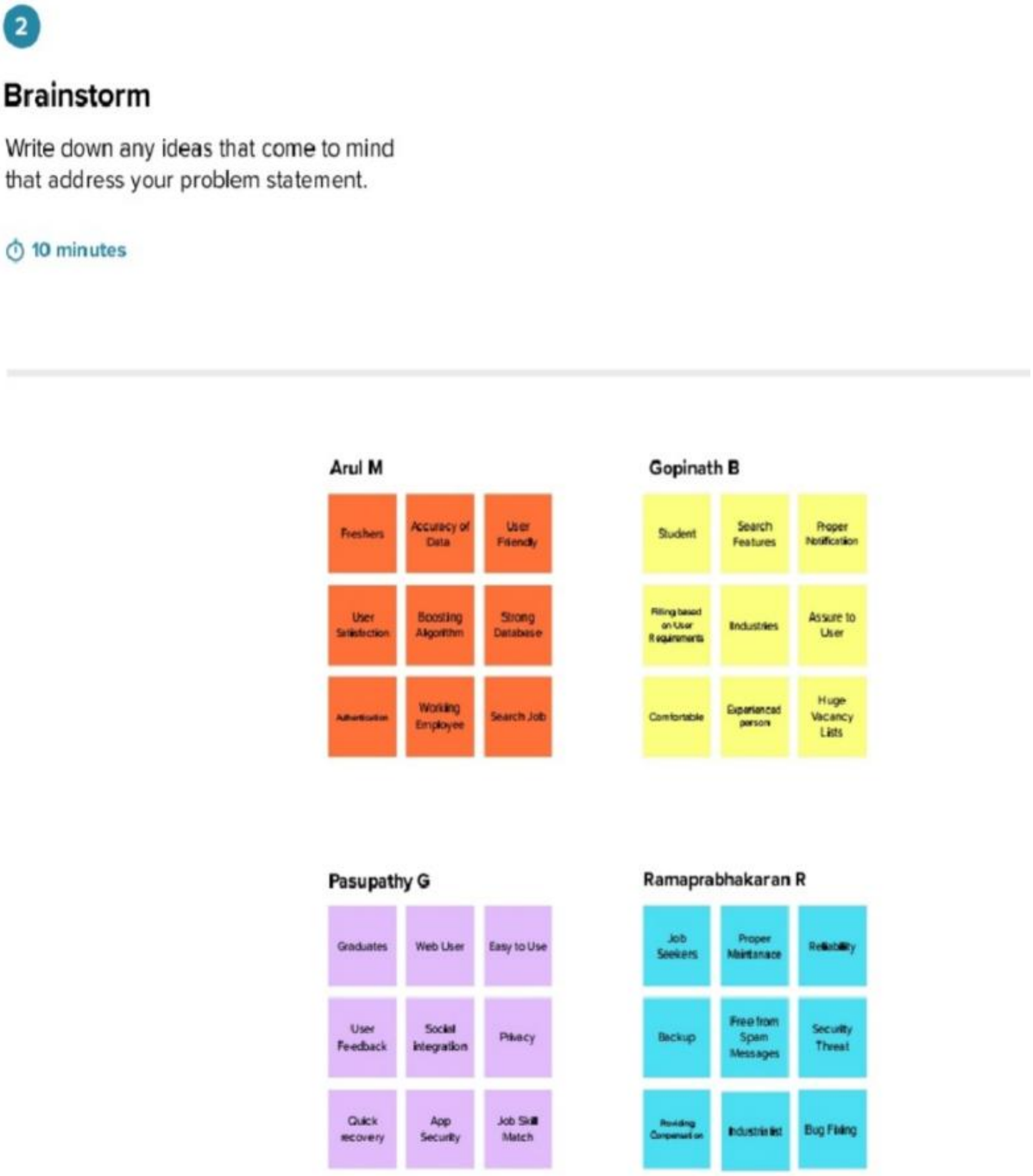
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads toproblemsolving. Prioritizing volume over value, out-of-the-box ideas are welcome andbuiltupon, and all participants are encouraged to collaborate, helping each other developa 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 thesameroom.

TEAM ID :PNT2022TMID46377

Step-1: Team Gathering, Collaboration and Select the ProblemStatement:

TEAM ID :PNT2022TMID46377

Step-2: Brainstorm, Idea Listing and Grouping:



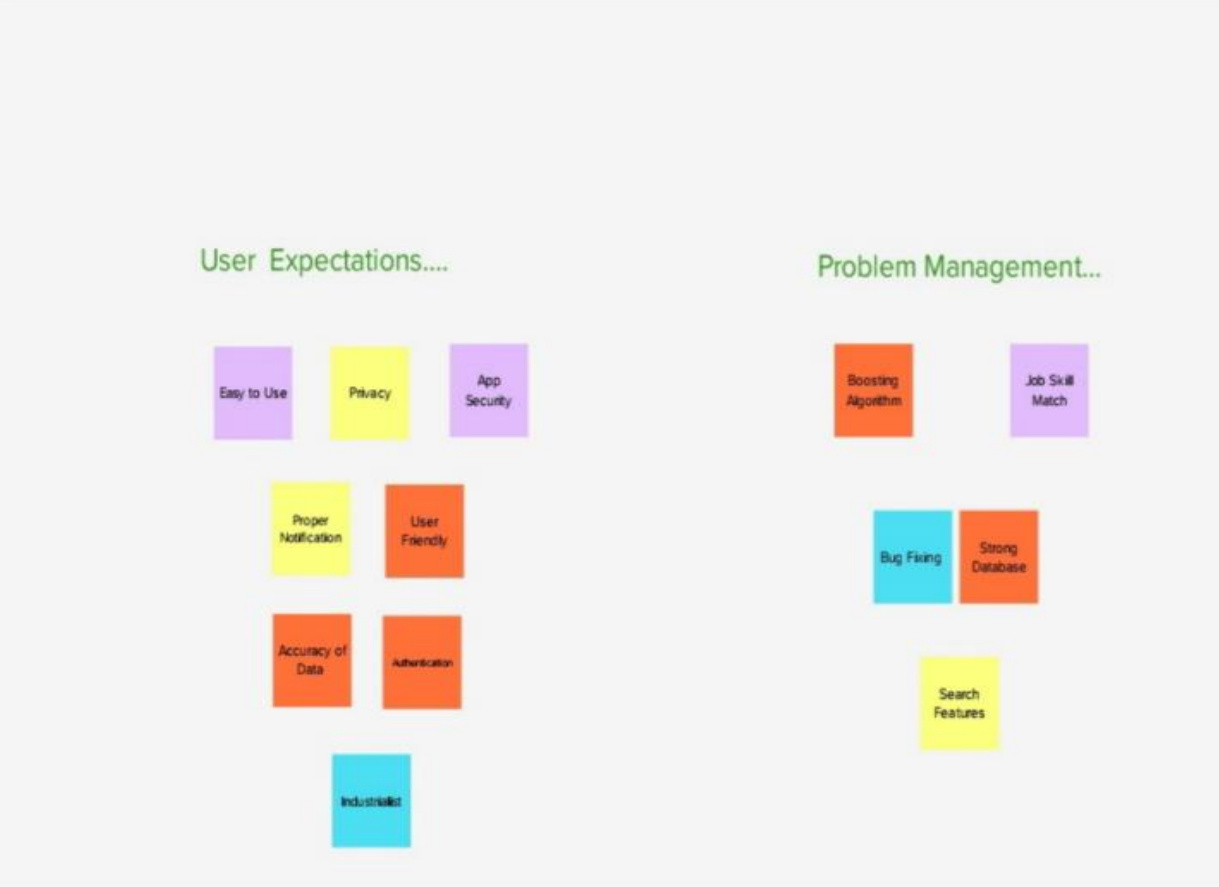
TEAM ID :PNT2022TMID46377

3

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 and break it up into smaller sub-groups.

20 minutes



TEAM ID :PNT2022TMID46377

Step-3: Idea Prioritization:

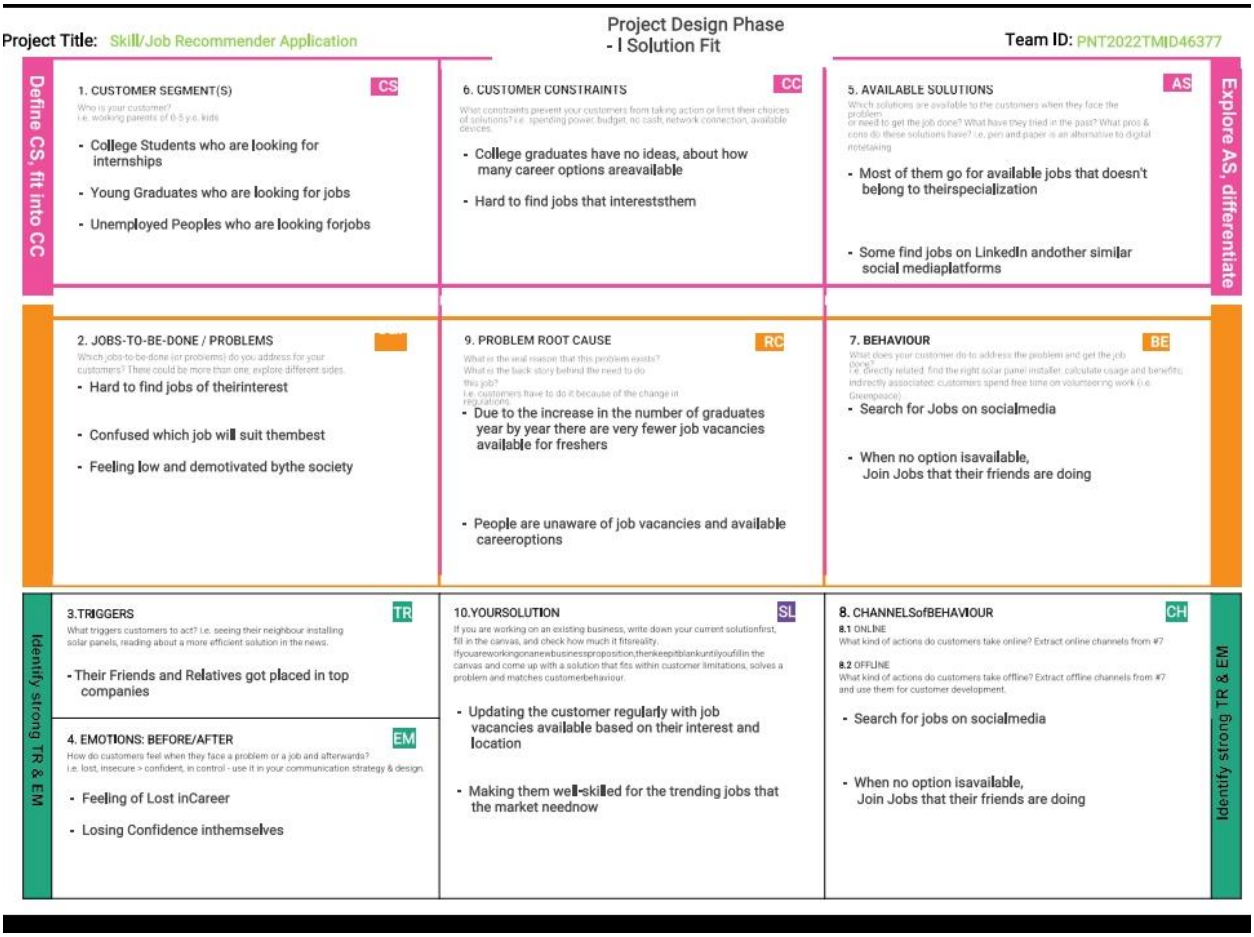
TEAM ID :PNT2022TMID46377

3.3 PROPOSED SOLUTION

SI.NO	*Parameter*	*Description*
01.	Problem Statement(Problem to be solved)	Premium policy is an issue to the users.Look for field based jobs as searching for fields as a whole is time-consuming.Estimating salaries based on technical skills.
02.	Idea/Solution Description	Free access to every users. Filtering job by it's Categories. Salary calculator for the estimation of the pay.
03.	Novelty / Uniqueness	Refinement of the job Fields. Earnings estimator based on knowledge of users.
04.	Social Impact/Customer Satisfaction	Open doors for every users as there is free access.
05.	Business Model(Revenue Model)	Advertising about the Platform. Regularly updating the new technologies and job offers.
06.	Scalability of the Solution	Scalable at Professional Training and Coaching. Scalability in finding more parent-friendly environment. Creating a positive culture is the main cause in maximizing the productivity.

TEAM ID :PNT2022TMID46377

3.4 PROBLEM SOLUTION FIT



04.REQUIREMENT ANALYSIS

4.1 Functional Requirements:

Following are the functional requirements of the proposed solution.

FR.NO	*Functional Requirement (Epic)*	*Sub Requirement (Story / Sub-Task)*
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login through Form Login through Gmail Login through LinkedIN
FR-4	User Profile	Updation of the user profile through the Login credentials
FR-5	User Search	Exploration of Jobs that users search for using the filters
FR-6	User Acceptance	Confirmation of the job

4.2 Non-functional Requirements:

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

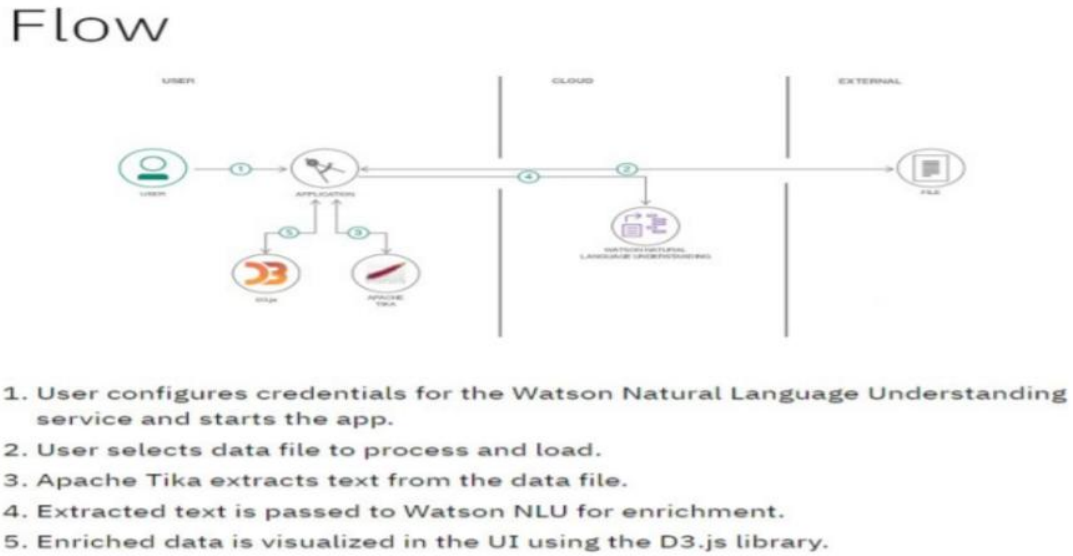
FR NO.	*Non-Functional Requirement*	*Description*
NFR-1	Usability	This application can be used by the job seekers to login and search for the jobs.
NFR-2	Security	This application is secure with the personalised Login credentials.

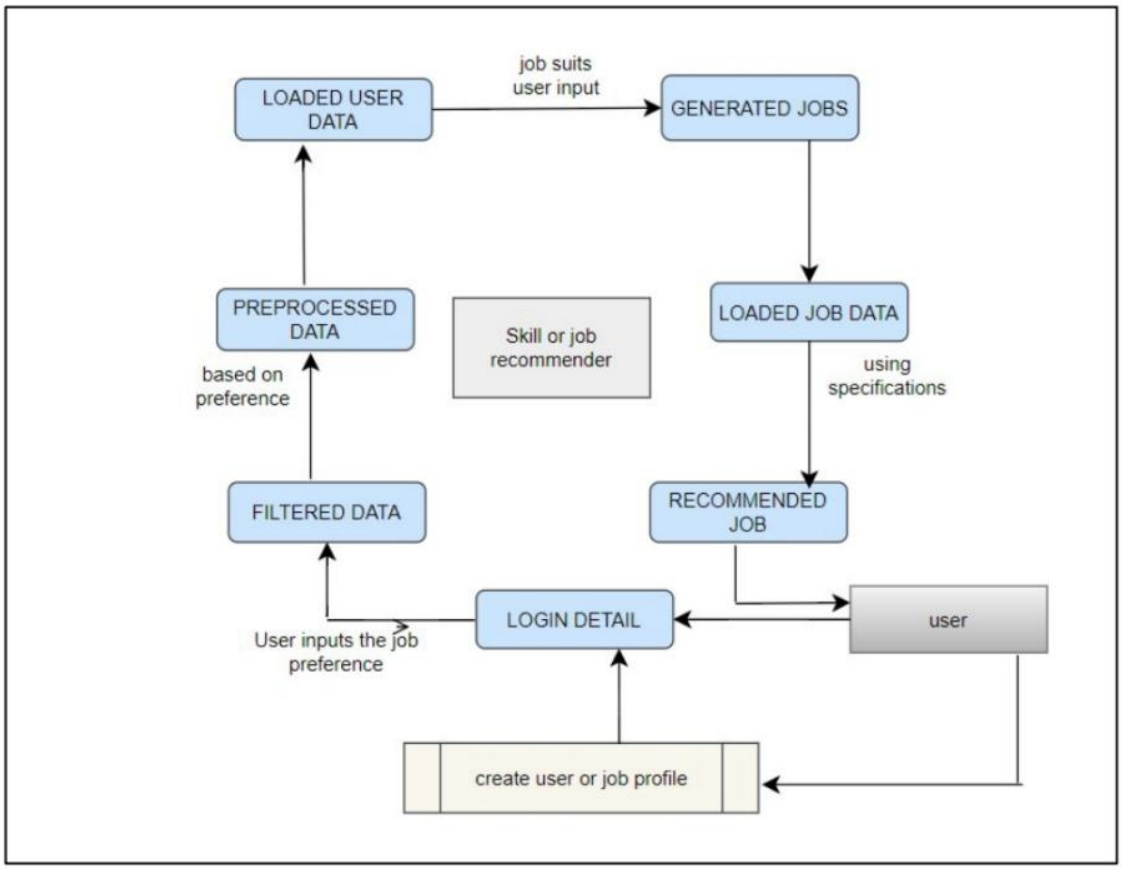
NFR-3	Reliability	This application is trustworthy which provides good job offers and suggestions of skill set with real time notifications.
NFR-4	Perfomance	The performance of the application is quicker responses to search of job seeker.
NFR-5	Availability	This application provides job offers and skillset with user's preference.
NFR-6	Scalability	The response time of the application is quite faster with user choice.

05.PROJECT DESIGN

5.1 Data Flow Diagrams:

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





5.2 Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

• Find the best tech solution to solve existing business problems.

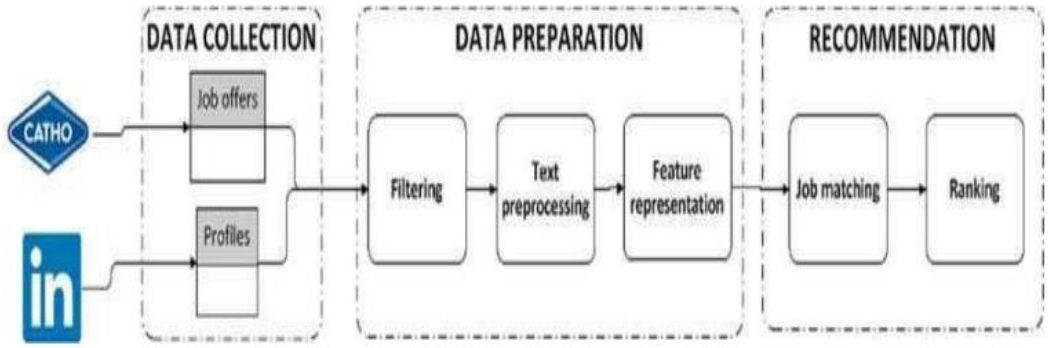
• Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.

• Define features, development phases, and solution requirements.

• Provide specifications according to which the solution is defined, managed, and delivered.

TEAM ID :PNT2022TMID46377

Solution Architecture Diagram:



5.3 User Stories:

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Job Seeker (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	As a user, I can access the dashboard for jobs to search	I can access the dashboard	High	Sprint-1
Job Seeker (Web user)	Login	USN-7	As a user, I can log into the application by entering email & password	I can log into the account I have created	High	Sprint-1
Administrator	Registration	USN-8	As a administrator, I can see the registration details	I can access the registration details	High	Sprint-1
	Login	USN-9	As a administrator, I can see the login details	I can access the login details	High	Sprint-1
	Dashboard	USN-10	As a administrator, I can update the dashboard	I can update the dashboard	High	Sprint-1

06.PROJECT PLANNING AND SCHEDULING

‘Project Planning and Scheduling’, though separate, are two sides of the same coin in project management. Fundamentally, ‘Project planning’ is all about choosing and designing effective policies and methodologies to attain project objectives. While ‘Project scheduling’ is a procedure of assigning tasks to get them completed by allocating appropriate resources within an estimated budget and time-frame. The basis of project planning is the entire project. Unlikely, project scheduling focuses only on the project-related tasks, the project start/end dates and project dependencies. Thus, a ‘project plan’ is a comprehensive document that contains the project aims, scope, costing, risks, and schedule. And a project schedule includes the estimated dates and sequential project tasks to be executed.

PROJECT PLANNING:

The project planning phase refers to:

- Developing a project to make it ready for investment
- Determines the jobs/tasks required to attain project objectives

6.1Sprint Planning & Estimation

What is sprint planning?

- Sprint planning is an event in scrum that kicks off the sprint

.

- The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.

- Sprint planning is done in collaboration with the whole scrum team. In scrum, the sprint is a set period of time where all the work is done.

- However, before you can leap into action you have to set up the sprint. You need to decide on how long the time box is going to be, the sprint goal, and where you're going to start.

- The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful.
- Bad sprint plans can derail the team by setting unrealistic expectations.

6.2 SPRINT DELIVERY SHEDULE

TEAM ID :PNT2022TMID46377

Sprint	Functional Requirement(Epic)	User Story Number	UserStory/Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user,I can register for the application by entering my email,password,and confirming my password.	5	High	Pasupathy.G Arul.M
Sprint-3	Register	USN-2	As a user register instantly using Gmail	4	Low	Gopinath.B
Sprint-1	Login	USN-3	Asauser,I can login to the application by entering myemail & password	5	High	Ramaprabhakaran
Sprint-1	Dashboard	USN-4	As a user I can access the dashboard there able to see jobs and filter the jobs using keywords.	6	High	Pasupathy.G
Sprint-3	Dashboard	USN-5	A dashboard which shows applied for jobs	6	Medium	Arul.M
Sprint-2	Profile	USN-6	As a user I can see my profile	4	Medium	Gopinath.B
Sprint-2	Profile	USN-7	As a user I can update my profile	4	Medium	Ramaprabh akaran.R
Sprint-1	Apply	USN-8	As a user view and apply for the job successfully	4	Medium	Arul.M
Sprint-3	Track	USN-9	track the status of the jobs through adash boardore mail services	4	Medium	Pasupathy G
Sprint-3	Email	USN-10	As a user get an email about new jobs	6	High	Pasupathy.G Arul.M
Sprint-2	Apply	USN-11	A user noticed after successfully applied job	6	Medium	Gobinath.B
Sprint-2	Bot	USN-12	A bot is embedded in the webpage it help to users instant matched skill jobs active	6	High	Pasupathy.G Arul.M
sprint-4	deploy	USN-13	Creating Docker image	5	Medium	Pasupathy.G Arul.M Gobinath.B Ramaprabhakaran.R
Sprint-4	UI	USN-14	Making Ui more interactive	5	Low	Pasupathy.G Arul.M Gobinath.B Ramaprabhakaran.R
sprint-4	IBM Container	USN-15	upload image to IBM container Registry	5	Medium	Pasupathy.G Arul.M Gobinath.B Ramaprabhakaran.R
sprint-4	Kubernetes	USN-16	Deploy on Kubernetes	5	Medium	Pasupathy.G Arul.M Gobinath.B Ramaprabhakaran.R

TEAM ID :PNT2022TMID46377

ProjectTracker,Velocity&BurndownChart:

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	29Oct2022
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

7.CODING & SOLUTIONING

7.1. FEATURE-1(SPRINT-1)

*REGISTER.HTML

*SIGNUP.HTML

REGISTER.HTML:

```
<html >

        <head>
          <title>Registration page</title>
          <link rel="stylesheet" href="{{url_for('static', filename='css/styles.css')}}">
        </head>
        <body>
          <header>
            <nav>
              <div class="logo">
                <p>SJR.COM</p>
              </div>
```

```
                <ul>
                    <li><a href="{{url_for('home')}}">Back to page</a></li>
                </ul>
            </nav>
        </header>


        <div class="container">
        <form action="/register" method="POST">

            <h2 style="text-align: center;">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>
            <!-- <input type="submit" class="submitbtn"value="submit" />
            <p>Already have a account <a href="{{url_for('signup')}}">Sign in</a></p> -->
            </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>
     <div style="text-align: center;"><a>
            <input type="submit" class="submitbtn"value="submit" />
            <p>Already have a account <a href="{{url_for('signup')}}">Sign in</a></p>
            </div>
```

```
        </form>
        </div>
      </body>

    <html>
```

SIGNUP.HTML:

```
<htm
l>
      <head>
        <title>Login page</title>
        <link rel="stylesheet" href="{{url_for('static', filename='css/styles.css')}}">
      </head>
      <body>
        <nav>
          <div class="logo">
            <p>SJR.COM</p>
          </div>
          <ul>
            <li><a href="{{url_for('home')}}">Back to page</a></li>
          </ul>
        </nav>


        <div class="container"> <br /><br />
        <form action="/signup" method="POST">
         <h1 style="text-align: center;">Login</h1> <br /><br />
          <label class="form_label"for="email"><b>Email</b></label><br><br>
          <input class="form_input"type="text" name= "email" /><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">
                  <ul>

                    <p>Don't have a account <a href="{{url_for('register')}}">Create new account</a></p>
                  </ul>
                </center>
              </form>
              </div>

            </body>
          </html>
```

7.2.FEATURE-2(SPRINT-2)

*APPLY JOB.HTML
*SERVER.PY

APPLY JOB.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">
<link rel="stylesheet" href="{{url_for('static', filename='css/testing.css')}}">
</head>
<body>
<nav>
<div class="logo">
<p>SJR.COM</p>
</div>
<ul>
<li><a href="/skill">SKILL</a></li>
<li><a href="{{url_for('home')}}">LOG OUT</a></li>
</ul>
</nav>
```

```
<section id="jobs">
<img style="position:absolute;top: 250px;"src="{{url_for('static',filename='php.jpg')}}"
alt=""width="250" height="250">
<img style="position:absolute;top: 600px;"src="{{url_for('static',filename='python.png')}}"
alt=""width="430" height="250">
<img style="position:absolute;top: 250px;left:
540px;"src="{{url_for('static',filename='software.jpg')}}" alt=""width="250" height="250">
<img style="position:absolute;top: 600px;left:
540px;"src="{{url_for('static',filename='java.jpg')}}" alt=""width="250" height="250">
<img style="position:absolute;top: 250px;left:
1050px;"src="{{url_for('static',filename='web.jpg')}}" alt=""width="250" height="250">
<img style="position:absolute;top: 600px;left:
1050px;"src="{{url_for('static',filename='sql.jpg')}}" alt=""width="250" height="250">
<h2 style="position:absolute;top: 500px;left:80px"><a
href="https://in.indeed.com/PHP-Developer-jobs?vjk=10cca9575b193c7d">APPLY</a></h2>
<h2 style="position:absolute;top: 500px;left:650px"><a
href="https://in.indeed.com/Software-Developerjobs?vjk=b7da08f07cac87d5">APPLY</a></h2>
<h2 style="position:absolute;top: 500px;left:1150px"><a href="https://in.indeed.com/WebDeveloper-
jobs?vjk=b81e49165da51eeb">APPLY</a></h2>
<h2 style="position:absolute;top: 850px;left:1150px"><a href="https://in.indeed.com/SQLDeveloper-
jobs?vjk=ac86b15908022123">APPLY</a></h2>
<h2 style="position:absolute;top: 850px;left:650px"><a href="https://in.indeed.com/JavaDeveloper-
jobs?vjk=da306a665e00eb30">APPLY</a></h2>
<h2 style="position:absolute;top: 850px;left:80px"><a href="https://in.indeed.com/PythonDeveloper-
jobs?vjk=fa7b9bd250044569">APPLY</a></h2>
</section>
</body>
</html>
```

SERVER.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__)
app.secret_key='a'
try:
```

```
  conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=824dfd4d-99de-440d-9991-
629c01b3832d.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30119;SECU
RITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=kcf08623;PWD=8himlHE
H6rDcSV2i",",")
except:
   print("Unable to connect: ",ibm_db.conn_error())
@app.route('/')
def home():
 return render_template('home.html')
@app.route("/register",methods=['GET','POST'])
def register():
   error = None
   if request.method=='POST':
         username=request.form['username']
         email=request.form['email']
         phonenumber=request.form['phonenumber']
         password=request.form['password']
         sql="SELECT * FROM user WHERE phonenumber=?"
         prep_stmt=ibm_db.prepare(conn,sql)
         ibm_db.bind_param(prep_stmt,1,phonenumber)
         ibm_db.execute(prep_stmt)

         account=ibm_db.fetch_assoc(prep_stmt)
         print(account)

SENDGRID_API_KEY='SG.syXUVAihRRuGI0DvhxY6Tw.eKTfa3dnL0yimAvWO9gYgoCoVw
K3-lN9TAGPi1UT0BM'
#
SG.29Td0tbNSkyliF9SSPnQNA.4DBECk8ka8RmmYRE5OIsRKGOR2QI2raRG3CLmdsVB
Vc
        message = Mail(
           from_email='skilljob007@gmail.com',
           to_emails=email,
           subject='Hello there! Welcome to Skill And Job Recommender',
           html_content='<strong>SJR warmly welcomes YOU!!!,Thanks for taking the time
to apply for our position.we appreciate your interest in SJR.COM</strong</strong>')
        try:
          sg =
```

```
SendGridAPIClient('SG.eablvkxWThCaGaY5zvBe6g._MsF4iOdsOaR0CBOmHK_TapO0o8
SQpnXRGNBjiCCs60')
            response = sg.send(message)
            print(response.status_code)
            print(response.body)
            print(response.headers)
          except Exception as e:
            print(str(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(prep_stmt,1,email)
             ibm_db.bind_param(prep_stmt,2,username)
             ibm_db.bind_param(prep_stmt,3,phonenumber)
             ibm_db.bind_param(prep_stmt,4,password)

             ibm_db.execute(prep_stmt)
             flash(" Registration successfull. Log in to continue !")
    else:
       pass
    return render_template('register.html',error=error)
@app.route('/signup',methods=['GET','POST'])
def signup():
    error = None
    if request.method=='POST':
       username=request.form['email']
       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['EMAIL']
          session["email"]=account["EMAIL"]
          flash("Logged in successfully!")
```

```
            return render_template('applyjob.html')
        else:
            error="Incorrect username / password"
            return render_template('signup.html',error=error)
    return render_template('signup.html',error=error)
@app.route('/applyjob')
def applyjob():
    return render_template('applyjob.html')
@app.route('/skill')
def skill():
  return render_template('skill.html')

@app.route('/aboutus')
def aboutus():
  return render_template('aboutus.html')
if __name__=='__main__':
   app.run(host='0.0.0.0',port=5000,debug=True)
```

8.TESTING

8.1 TESTCases

TestcaseID	FeatureType	Component	TestScenario
LoginPage_TC_O O1	Functional	HomePage	Verifyuser is able to see theLogin/Signup popup when userclickedonMy accountbutton
LoginPage_TC_O O2	UI	HomePage	Verify the UI elements inLogin/Signuppop up
LoginPage_TC_O O3	Functional	Home page	Verify user is able to log intoapplicationwithV alidcredenti als
LoginPage_TC_O O4	Functiona	Loginpage	Verify user is able to log intoapplicationwithI nValidcredenti also
LoginPage_TC_O O5	Functional	Loginpage	Verify user is able to log intoapplicationwithI

			nValidcredenti als

SI.no	**Steps To Execute**	**TestData**
1	1.Enter URL and click go. 2.Click on My Account dropdown button. 3.Verify login/Singup popup displayed or not.	index.html
2	1.Enter URL and click go. 2.Click on My Account dropdown button. 3.Verify login/Singup popup with below UI elements: a.email text box b.password text box c.Login button d.New customer? Create account linke.Last password? Recovery	index.html
3	1.Enter URL(index.html) and click go. 2.Click on My Account dropdown button. 3.Enter Valid username/email in Email textbox. 4.Entervalidpasswordinpas sword textbox. 5.Click on login button.	Username:pasupathy password:21082001
4	1.Enter URL(index.html) and click go. 2.Click on My Account dropdown button. 3.Enter InValid username/email in Email text box. 4.Enter valid password in password textbox.	Username:pasupathy password:21082001

	5..Click on login button.	
5	1.Enter URL(index.html) and click go. 2.Click on My Account dropdown button. 3.Enter Valid username/email in Email textbox. 4.Enter valid password in as word textbox. 5..Click on log in button.	Username:pasupathy password:21082001
6	1.Enter URL(index.html) and click go. 2.Click on My Account dropdown button. 3.Enter Valid username/email in Email textbox. 4.Enter valid password in password textbox. 5..Click on login button.	Username:pasupathy password:21082001

ExpectedResult	**ActualResult**	**Status**
Login/Signup popup should display	Working as expected	pass
Application should show below UI elements: a.email text box b.password text box c.Login button with orange colourd.New customer? Create account link. Last password	Working as expected	pass
User should navigate to user account homepage	Working as expected	pass
Application should show 'Incorrect email or password ' validati on message.	Working as expected	pass
Application should show 'Incorrect email or	Working as expected	pass

password ' validation message.		
Application should show 'Incorrectemail or password ' validation message	Working as expected	pass

8.2.USER ACCEPTANCE TESTING: Purpose of Document:

Purpose of Document:

The purpose of this document is to briefly explain the test coverage and open issues of the [Product Name] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

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

Resolution	**Severity1**	**Severity2**	**Severity3**	**Severity4**	**Subtotal**
Subtotal	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Repduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	27	14	13	26	77

TEAM ID :PNT2022TMID46377

Test Case Analysis:

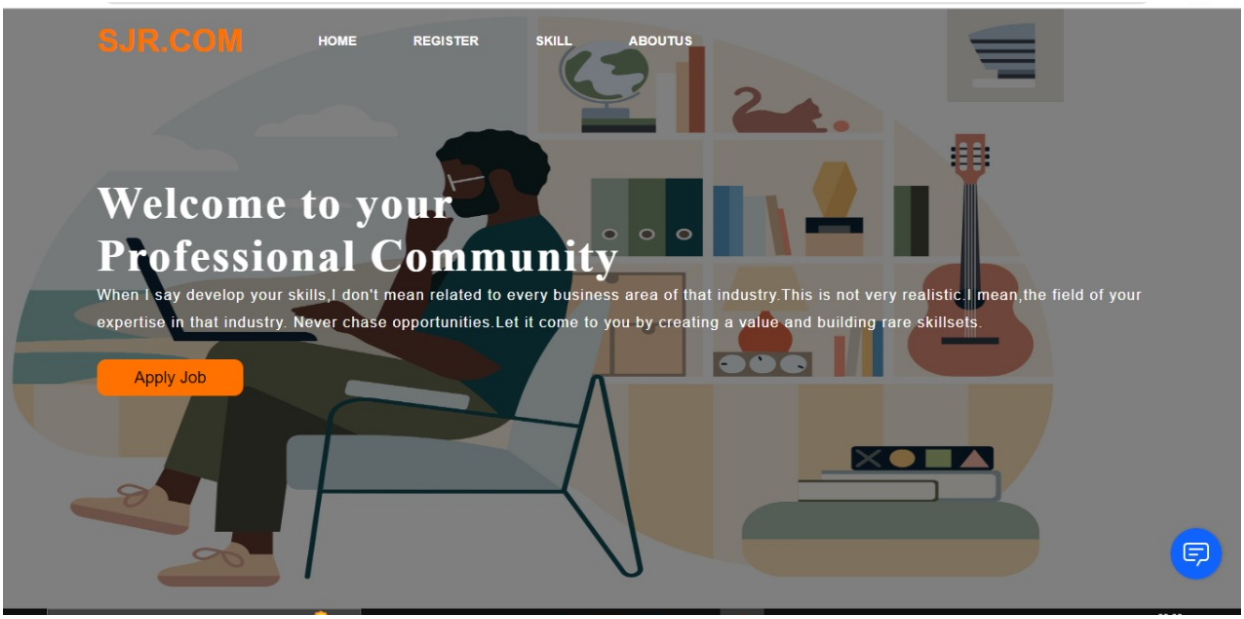
This report shows the number of test cases that have passed,failed,and untested

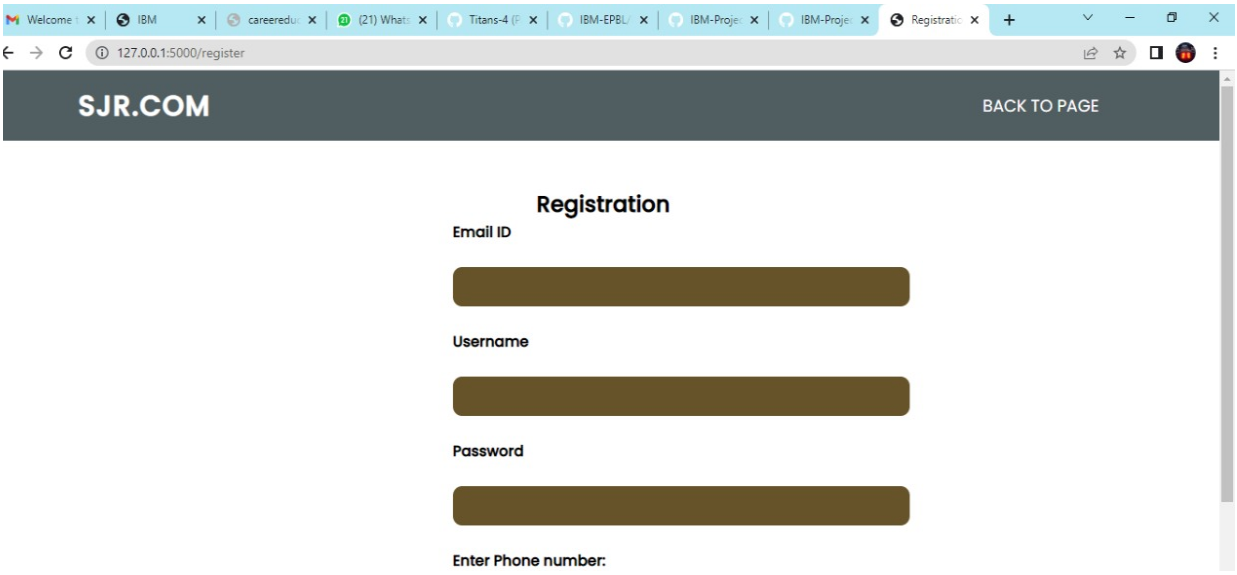
Resoluion	**Total Cases**	**Not Tesed**	**Fail**	**Pass**
Print Enne	7	0	0	7
Client Aplication	51	0	0	51
Security	2	0	0	2
Out source Shipping	3	0	0	30
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

TEAM ID :PNT2022TMID46377

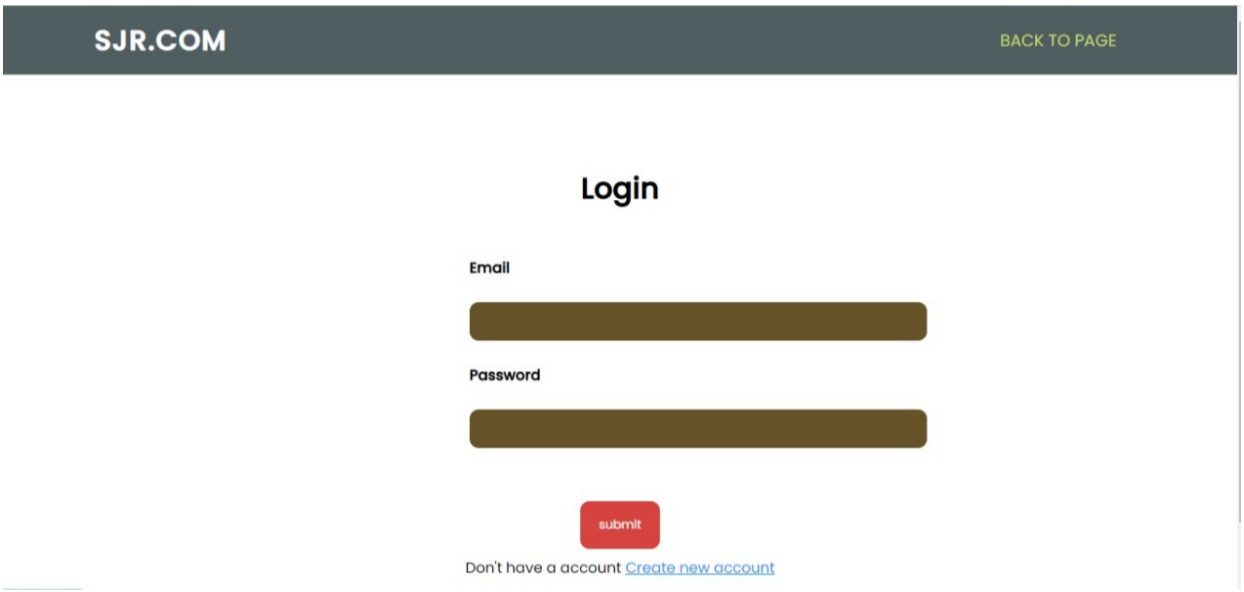
09.RESULTS

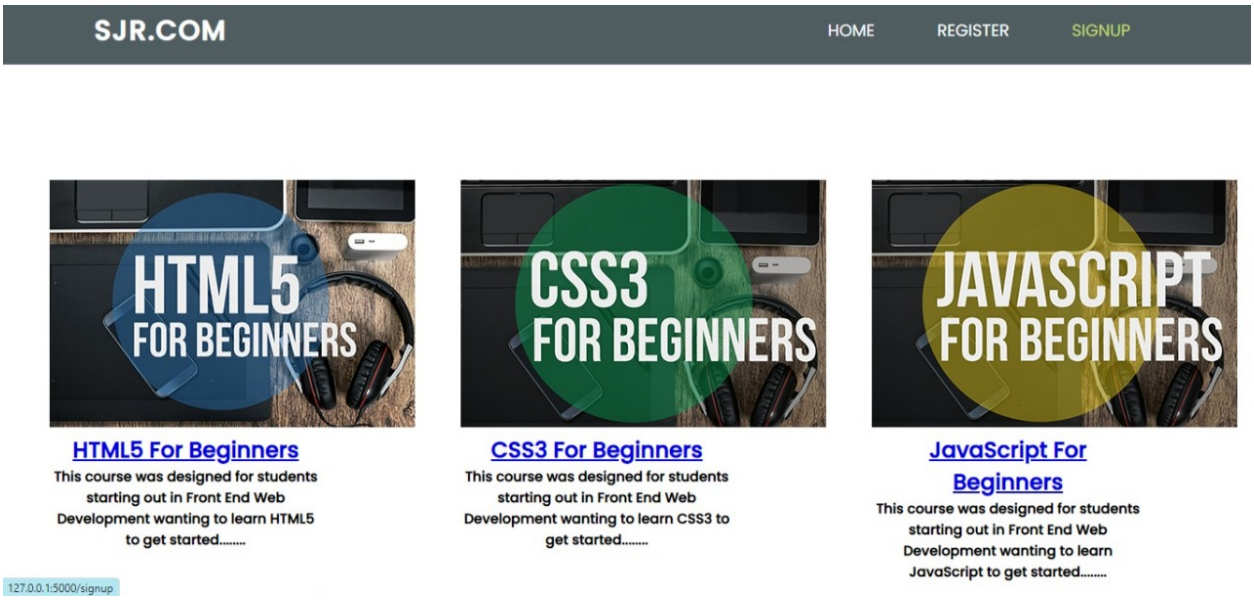
9.1 PERFOMANCE METRICS:





TEAM ID :PNT2022TMID46377





10.ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

- Bidirectional recommendation.Relational aspects are included.

- Adaptive system.
- Use many attributes.
- Use ontology to categorize jobs and as a knowledge base to define features.

DISADVANTAGES:

- Less attributes used.
- No perfect measures.
- Knowledge acquisition and knowledge engineering problems.

11.CONCLUSION:

Job Recommendation System has a major role to play among recommending systems. With the presence of new algorithms and techniques, the system needs to evolve along with it. The main objective of this project is to recommend a suitable job for the candidates. This project has two pre-processing methods, one text mining method and one similarity function. The pre-processing methods are stop words and porter stemmer.

The text mining method is tf-idf. The similarity function is a cosine similarity function. Pre-processing methods are used with resumes and with jobs description, to make the system more efficient by avoiding some garbage words. Tf-idif is used in processed resumes and processed jobs descriptions to convert it from text to matrix to compare. Cosine Similarity will measure the similarity between the resume and each job description.Finally, it will display the scores for the jobs in a sorted way. There is also a pie chart which is used to visualize the percentage of the scores which is got by the candidate for the jobs.

12.FUTURE SCOPE:

The objective of Skill and Job recommender systems is to **provide recommendations based on recorded information on the users' preferences**. These systems use information filtering techniques to process information and provide the user with potentially more relevant items.

13.REFERENCES:

[1] R. J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," in In Proceedings of DL ’00:Proceedings of the Fifth ACM Conference on Digital Libraries, New York,NY, pp. 13-20, 2000.

[2] Li-Ping Jing, Hou-Kuan Huang, Hong-Bo Shi, “Improved feature selection approach TFIDF in text mining”, International Conference on Machine Learning and Cybernetics, pp. 944-946, 2002, doi:10.1109/icmlc.2002.1174522.

[3] Shouning Qu ,Sujuan Wang,Yan Zou, ” Improvement of Text Feature Selection Method Based on TFIDF”, International Seminar on Future Information Technology and Management Engineering, pp. 79-81, 2008,doi:10.1109/fitme.2008.25.

[4] I. A. Braga, "Evaluation of stopwords removal on the statistical approach for automatic term extraction," Seventh Brazilian Symposium in Information and Human Language Technology, pp. 142-149, 2009.

[5] Nikolaos D. Almalis, Prof. George A. Tsihrintzis, Nikolaos Karagiannis, Aggeliki D. based job recommendation algorithm for job seeking and recruiting”, 6th International Conference on Information, Intelligence,Systems and Applications (IISA), pp. 1-7, 2015, doi:10.1109/iisa.2015.7388018.

[6] Mohammad Alodadi and Vandana P. Janeja, “ Similarity in Patient Support Forums Using TF-IDF and Cosine Similarity Metrics”,International Conference on Healthcare Informatics, pp. 521-522, 2015, doi:10.1109/ichi.2015.99.

[7] L. Zahrotun, “Comparison jaccard similarity, cosine similarity and combined both of the data clustering with shared nearest neighbor method,” Computer Engineering and Applications Journal. vol. 5. Pp. 11-18, 2016, doi:10.18495/comengapp.v5i1.160, 2016.

[8] Peng Yi, Cheng Yang ,Chen Li, Yingya Zhang, “A Job Recommendation Method Optimized by Position Descriptions and Resume Information”, IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), pp. 762 -764, March 2017,doi:10.1109/rteict.2017.8256590.

[9] Minh-Luan Tran, Anh-Tuyen Nguyen, Quoc-Dung Nguyen, Tin Huynh, “A comparison study for job recommendation”, International Conference on Information and Communications (ICIC), pp. 199-204, 2017,doi:10.1109/infoc.2017.8001667.

[10] Gokul P.P, Akhil BK, Shiva Kumar K.M, “Sentence similarity detection in Malayalam language using cosine similarity”, 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 221-225, 2017, doi:10.1109/rteict.2017.8256590.

14.APPENDIX:

14.1. SOURCE CODE: **LINK:** https://github.com/Titans-4

14.2:GITHUB & PROJECT LINK:

GITHUB ID: IBM-EPBL/IBM-Project-44252-1660723478