

SKILL / JOB RECOMMENDER APPLICATION

Bachelor of Engineering
In
Computer Science and Engineering

Submitted by

TEAMI ID: PNT2022TMID48837

T. VIJAYAMANIKANDAN (921019104057)

M. DINESH KUMAR (921019104701)

V. RAMANAN (921019104302)

K. VISHNU BALA (921019104058)

**NADAR SARASWATHI COLLEGE OF ENGINEERING ANDTECHNOLOGY,
THENI-625531**

ANNA UNIVERSITY: CHENNAI

TABLE OF CONTENT

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

3.2 Ideation & Brainstorming

3.3 Proposed Solution

3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement

4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams

5.2 Solution & Technical Architecture

5.3 User Stories

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

6.2 Sprint Delivery Schedule

7. CODING & SOLUTIONING (Explain the features added in the project along with code)

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

13.1 Source Code

13.2 GitHub & Project Demo Link

1.INTRODUCTION

1.1 OVERVIEW

Finding jobs have always been hard - from not having proper knowledge on the organization's objective, their work culture and current job openings to finding the right candidate with desired qualifications to fill their current job openings. Online Job Search Portals have since then been introduced making job seeking convenient on both sides. Job Portal is the solution where recruiter as well as the job seeker meet aiming at fulfilling their individual requirement. They are the cheapest as well as the fastest source of communication reaching a wide range of audience on just a single click irrespective of their geographical distance.

1.2 PURPOSE

Even though online job portals have existed for a while, they only brought in more challenges, like:

- The education system does not always fulfil and focus on individual person skill development.
- Spending hours to find useful info from enormous amount of posts online.
- People who lack industry knowledge are unclear about what exactly they need to learn in order to get a suitable job for them.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

- Job portals pretty much work using resume information to match people instead on customizing on a job seeker's skill set.
- Recruiters see very similar resumes of hundreds of applicants making it impossible to figure out which candidate seems to be relevant or better for the job at hand.

2.2 PROBLEM STATEMENT DEFINITION

The current problem recruitment is done manually, most available jobs in Nigeria can only be applied at the agency can be done for which job seekers have to go to the agency check the available jobs at the agency.

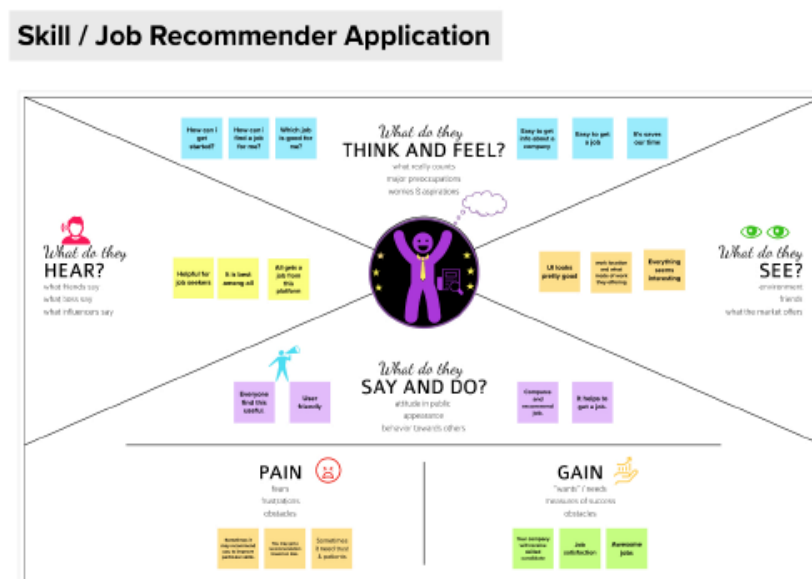
2.3 REFERENCES

- <https://ieeexplore.ieee.org/document/7944917>
- https://www.researchgate.net/publication/325697854Job_Recommendation_based_on_Job_Seeker_Skills_An_Empirical_Study
- <https://www.quora.com/LinkedIn>
- <https://ieeexplore.ieee.org/document/8960231>
- <https://ieeexplore.ieee.org/document/9752295>

3. IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

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

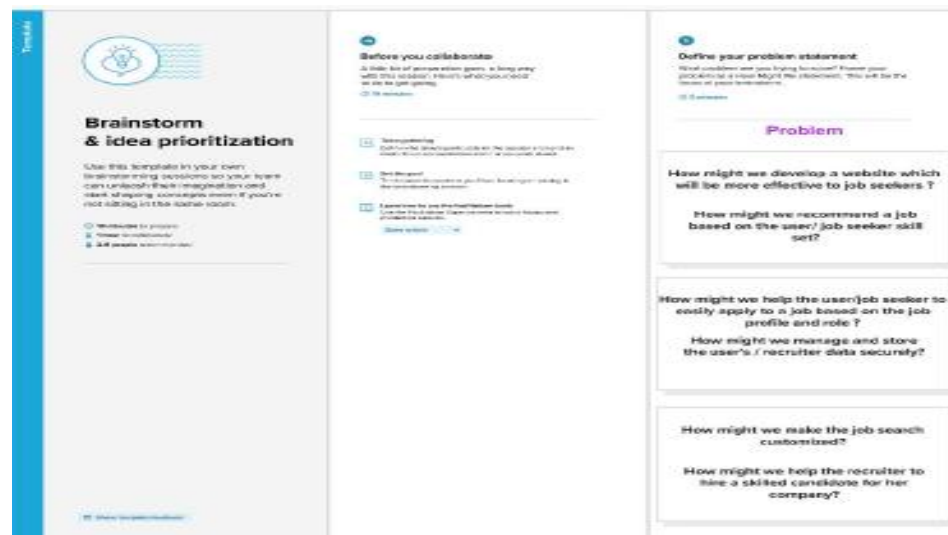


3.2 IDEATION AND BRAINSTORMING

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.

Step-1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & Idea prioritization

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.

Before you collaborate

- 1. Define the problem statement
- 2. Define the problem statement
- 3. Define the problem statement

Define your problem statement

What problem are you trying to solve? Please provide a clear and concise statement. This will be the focus of your brainstorming.

Problem

How might we develop a website which will be more effective to job seekers?

How might we recommend a job based on the user's job seeker skill set?

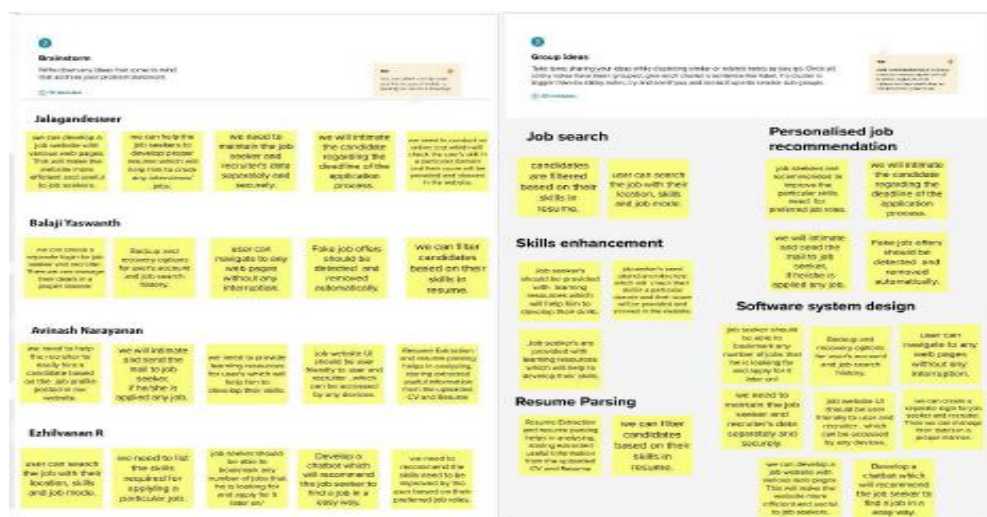
How might we help the user/job seeker to easily apply to a job based on the job profile and role?

How might we manage and store the user's / recruiter data securely?

How might we make the job search customized?

How might we help the recruiter to hire a skilled candidate for her company?

Step-2: Brainstorm, Idea Listing and Grouping



Brainstorm

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.

Group ideas

Take time to group your ideas into clusters or themes based on topics. Once you have grouped your ideas, you can then use the template to group your ideas into clusters or themes based on topics.

Job search

Candidates are filtered based on their skills and experience.

User can search the job with their location, skills and job mode.

Personalised job recommendation

Job seekers can get personalised job recommendations based on their skills and experience.

We will recommend the candidate based on the skills and experience.

Skills enhancement

Job seekers can get personalised job recommendations based on their skills and experience.

Job seekers can get personalised job recommendations based on their skills and experience.

Software system design

Job seekers can get personalised job recommendations based on their skills and experience.

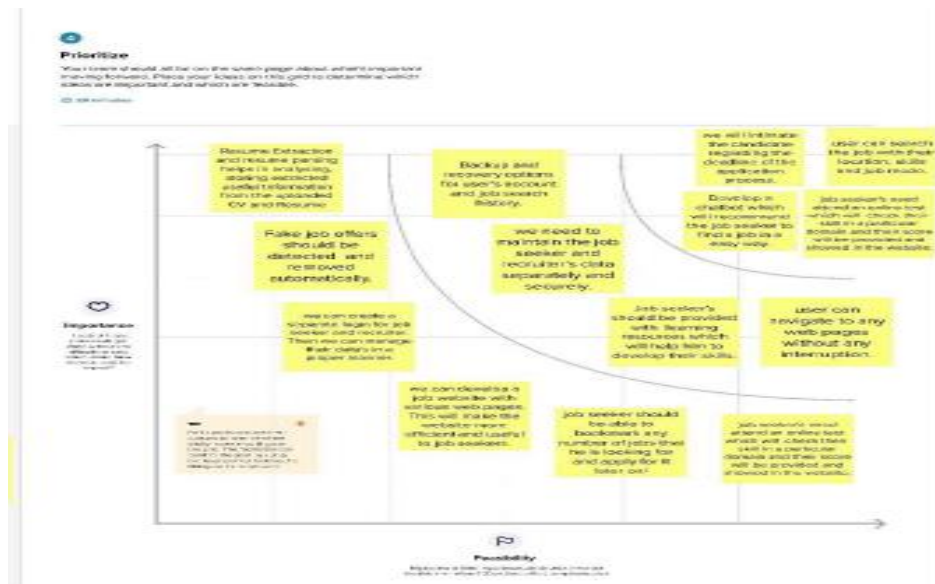
Job seekers can get personalised job recommendations based on their skills and experience.

Resume Parsing

Resume Parsing can be used to extract the skills and experience from the resume.

Resume Parsing can be used to extract the skills and experience from the resume.

Step 3: Idea Prioritization



3.3 PROPOSED SOLUTION

The system customizes and only shows recommended jobs based on the user's skill set and preferences (Using graphql api) Similarly, the same recommendation system helps provide job applicant recommendations to the job recruiters to find the most eligible candidates for their firm. All important data - job seeker's and hoster's personal information needs to be also stored safely and securely. Using a sql database is the most easiest, safest and convenient way possible. Data needs to also be private in some cases like when information is shared with the host while applying for a job.

3.4 PROBLEM SOLUTION FIT

Define CS, MS and VC	1. CUSTOMER SEGMENT(S) <ul style="list-style-type: none"> Job Seeker Job Recommender 	6. CUSTOMER CONSTRAINTS <ul style="list-style-type: none"> Lack of awareness about a job openings Personal data security Vulnerable to employment scams 	5. AVAILABLE SOLUTIONS <ul style="list-style-type: none"> Linked in, indeed, and Naukri are some of the leading sources for job opportunities. They intimate user (job seeker) with a notification about a recent job openings based on their skills Premium user will get more features including learning resources, etc. 	Define CS, MS and VC
	2. JOBS-TO-BE-DONE / PROBLEMS <p>Job Seeker:</p> <ul style="list-style-type: none"> Finding desired job is not an easy task. They need to gain knowledge before applying a particular job They should be aware of fraudulent job post <p>Job Recruiter:</p> <ul style="list-style-type: none"> They need to find a skilled candidate for her company The hiring process takes so much time to complete Filtering candidates is difficult 	9. PROBLEM ROOT CAUSE <ul style="list-style-type: none"> Increasing in population as well as increasing in graduates on particular domain leads to Job Crisis The education system does not fully and focus on individual person skill development 	7. BEHAVIOUR <ul style="list-style-type: none"> Learn and see more about a job openings in job posting website Develop and improve her knowledge Connect with recruiters on linked in platform and maintain a friendly connection with people 	Define CS, MS and VC
Identify driving MS & VC	3. TRIGGERS <ul style="list-style-type: none"> Financial Problem Societal pressure Dissatisfaction of job Find a better way to improve her knowledge as well as career growth 	10. YOUR SOLUTION <ul style="list-style-type: none"> A Fake Job Offer is detected and removed automatically Recommend a skill to job seeker for a particular job openings A notification will be sent via email regarding job openings Learning resources will be provided, then it will improve the user knowledge and skills 	8. CHANNELS of BEHAVIOUR <p>ONLINE</p> <ul style="list-style-type: none"> Apply and maintain a connection with recruiters Also search about job openings <p>OFFLINE</p> <ul style="list-style-type: none"> Learn and gain the required skills in open Source platform as well as in our Job Website 	Define CS, MS and VC
	4. EMOTIONS: <p>BEFORE</p> <ul style="list-style-type: none"> Sad, depressed, and low confidence Fear of Rejection before attending any hiring process <p>AFTER</p> <ul style="list-style-type: none"> Highly Motivated Gained confidence to do any task 			Define CS, MS and VC

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

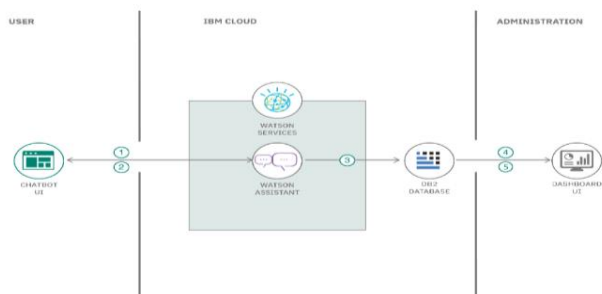
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Chat Bot	A Chat Bot will be there in website to solve user queries and problems related to applying a job, search for a job and much more.
FR-4	User Login	Login through Form Login through Gmail
FR-5	User Search	Exploration of Jobs based on job filters and skill recommendations.
FR-6	User Profile	Updation of the user profile through the login credentials
FR-7	User Acceptance	Confirmation of the Job.

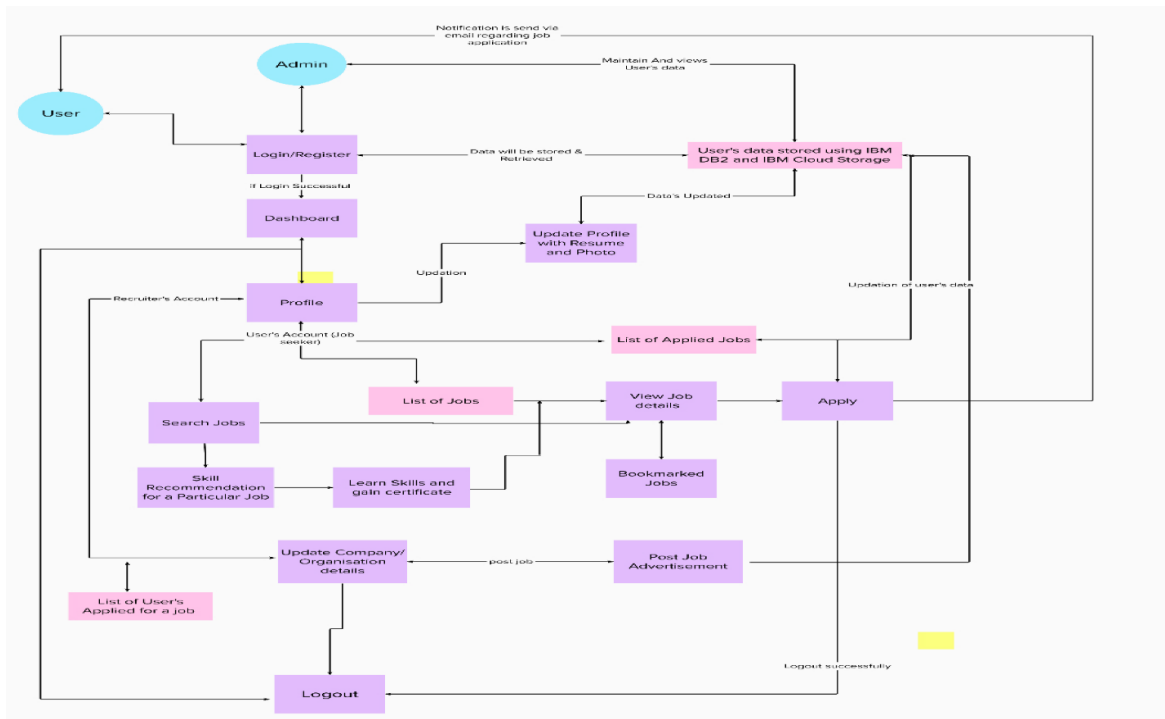
4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This application can be used by the job seekers to login and search for the job based on her Skills set.
NFR-2	Security	This application is secure with separate login for Job Seekers as well as Job Recruiters.
NFR-3	Reliability	This application is open-source and feel free to use, without need to pay anything. The enormous job openings will be provided to all the job seekers without any limitation.
NFR-4	Performance	The performance of this application is quicker response and takes lesser time to do any process.
NFR-5	Availability	This application provides job offers and recommends Skills for a Particular Job openings.
NFR-6	Scalability	The Response time of the application is quite faster compared to any other application.

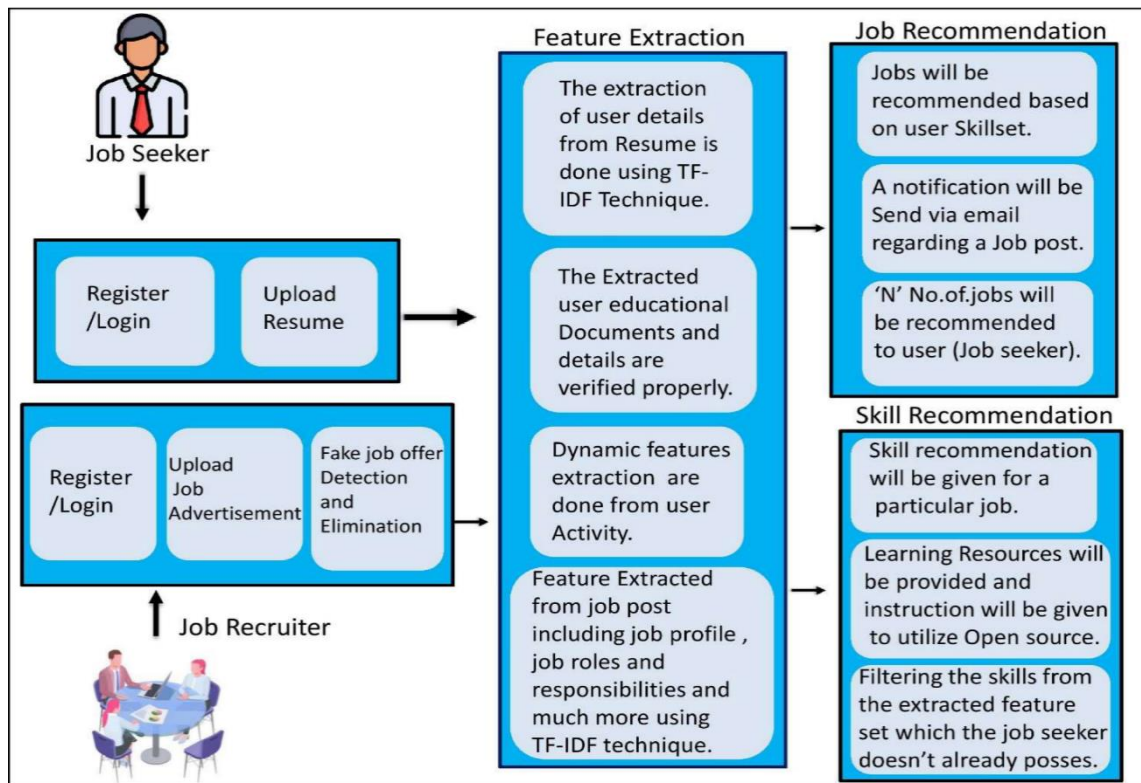
5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS





5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming 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 online websites	I can register & access the dashboard with online website Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation Gmail & click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	High	Sprint-1
	Dashboard					
Customer (Web user)		USN-6	As a user, I can able to take up the skill assessment and view the appropriate test score. Based on the skill sets I can able to get personalised job recommendations.	I can receive job recommendations	High	Sprint-1
Customer Care Executive		USN-7	As a customer care executive, we provide 24/7 chatbot support.	24/7 chatbot support	High	Sprint-1
Administrator		USN-8	As an administrator, I can able to view the progress and make required changes in the project	Deploy user specific and personalised job recommendations	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

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

6.2 SPRINT DELIVERABLE PLAN

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	UI Creation Creating Registration page, Login page	10	Medium	Dineshkumar Vijayamanikandan Ramanan Vishnubala
Sprint-1	Database Connectivity	USN-2	Viewing and applying jobs Connecting UI with Database	10	High	Vijayamanikandan Ramanan
Sprint-2	SendGrid Integration	USN-3	SendGrid Integration with Python Code	10	Low	Dineshkumar Vishnubala
Sprint-2	Chatbot Development	USN-4	Building a chatbot	10	High	Vijayamanikandan Vishnubala

Sprint-3	Integration and Containerisation	USN-5	Integrating chatbot to the HTML page and containerizing the app.	20	Medium	Vijayamanikandan Ramanan Vishnubala Dineshkumar
Sprint-4	Upload Image and deployment	USN-6	Upload the image to the IBM Registry and deploy it in the Kubernetes Cluster.	20	High	Dineshkumar Vishnubala

7. CODING AND SOLUTIONING

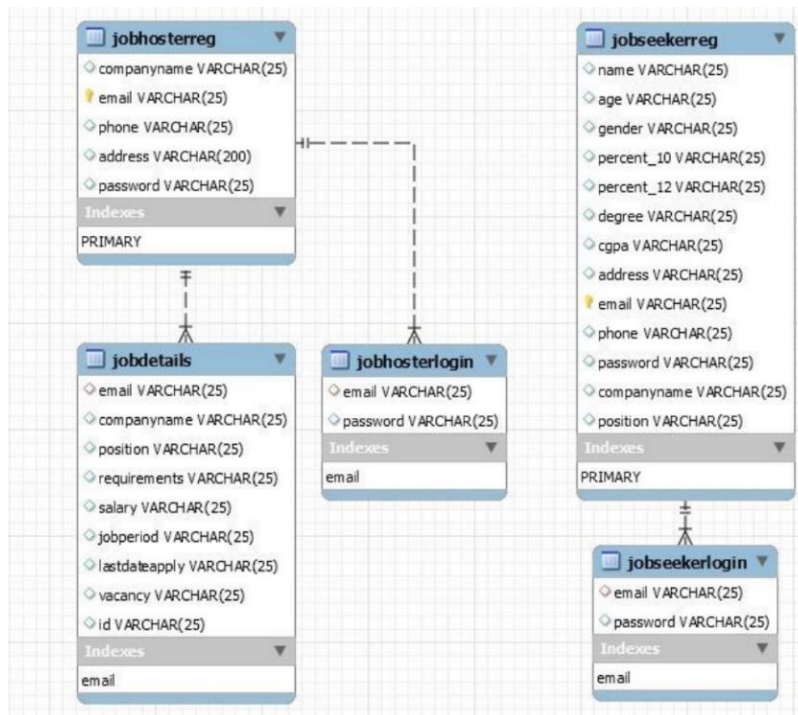
7.1 FEATURE 1

Skill based job recommendation – Jobs are recommended based on job seeker’s individual skill set. This also brings in custom list of jobs that’s different for different job seekers.

7.2 FEATURE 2

Hosting jobs– Job hoster can easily host jobs that can be accessed by a varied range of applicants. Additional feature – filtering through jobs based on skill, location, salary/stipend, mode of job (for both applying and hosting jobs).

7.3 DATABASE SCHEMA



8. TESTING

REGISTER PAGE

Back to page

Job Recommender

Registration

Email ID

Username

Password

Enter Phone number:

Enter four digit pin:

submit

Already have an account [Sign in](#)

Activate Windows
Go to Settings to activate Windows.

LOGIN PAGE

Back to page

Job Recommender

Login

Username

Password

submit

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

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

Activate Windows
Go to Settings to activate Windows.

[Back to page](#)

Job Recommender

Try to login with your 4 digit pin

Username

Pin

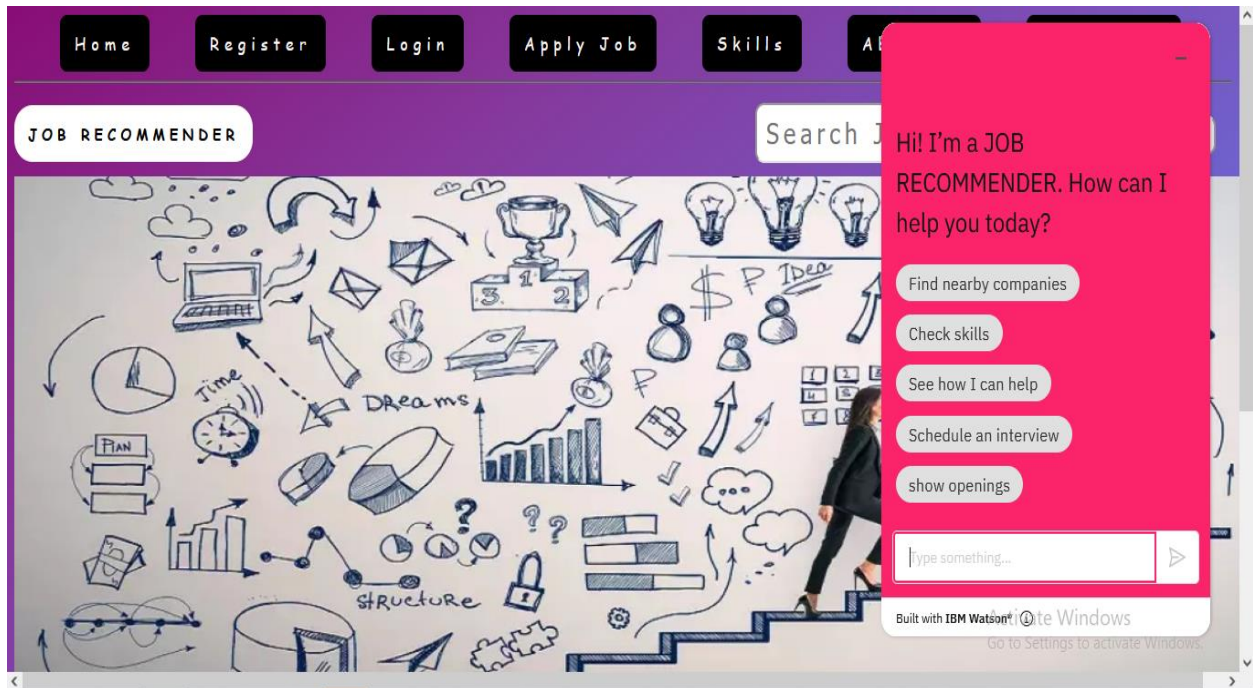
submit

[Back to login](#)

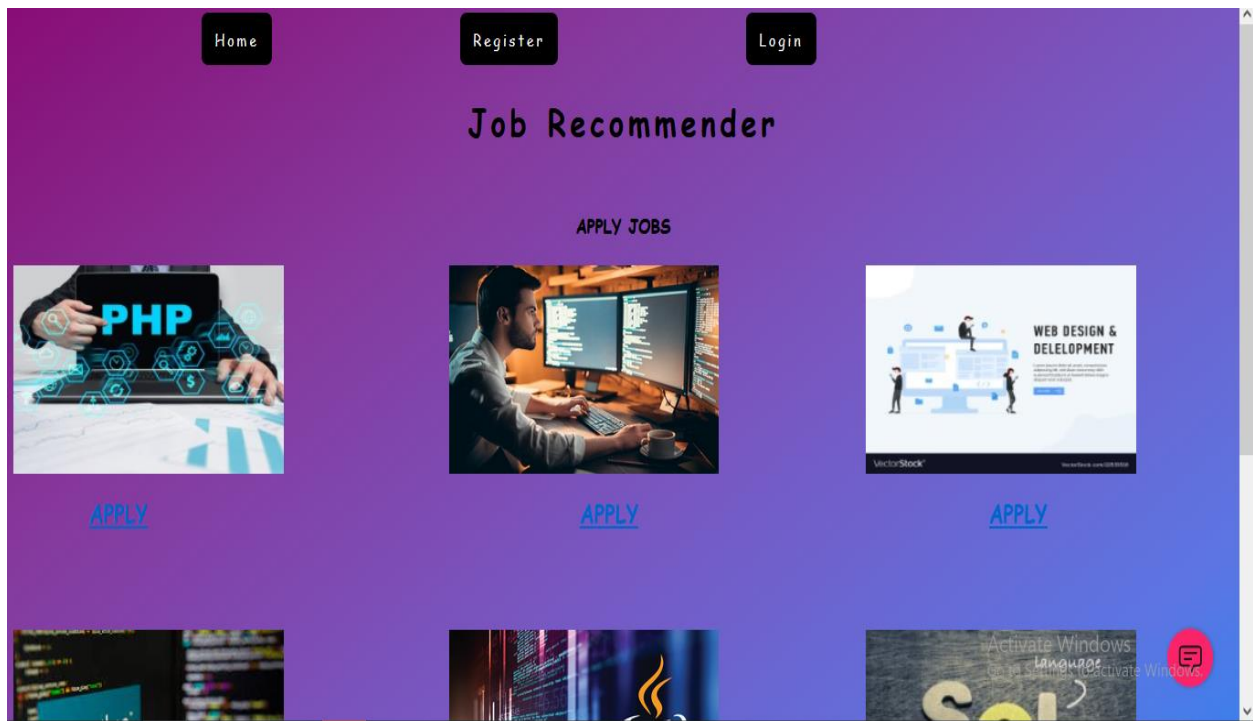
Activate Windows
Go to Settings to activate Windows.

[illegible]

CHATBOT PAGE



APPLY PAGE



9. RESULTS

9.1 PERFORMANCE METRICS

Based on the person-job fit premise, we propose a framework for job recommendation based on professional skills of job seekers. We automatically extracted the skills from the job seeker profiles using a variety of text processing techniques. Therefore, we perform the job recommendation using TF-IDF and four different configurations of Word2vec over a dataset of job seeker profiles and job vacancies collected by us. Our experimental results show the performances of the evaluated methods and configurations and can be used as a guide to choose the most suitable method and configuration for job recommendation.

10. ADVANTAGES AND DISADVANTAGES

- Sourcing candidates requires a lot of effort, which means it can cost a company both time and money. It was found in one study that referred candidates are 55% faster to hire, compared with employees sourced through career sites. An advantage of employee referrals is that your current team member makes the connection and saves the recruiter that initial time of sourcing the candidate. Further, the candidate could be a better match compared to other candidates who apply externally. This will also help expedite the process and cut back on the need to find alternative options.
- Employees will want to work with someone who will improve their own output and day-to-day workload. So, in most cases, you can have more confidence in the candidate's ability to perform the necessary tasks. Further, according to research done by Zao, nearly three in ten employers have caught a fake reference on an application. So, a personal recommendation that is already within the company can instill confidence that the reference is in fact valid and reputable.
- After two years, retention of referred employees is 45% compared to 20% from job boards. Employee referrals tend to stay around longer, perhaps because they are personally connected to their peers. That's not to mention that the referrer themselves may feel more respected and valued too after their company takes their recommendation. And when an employee feels respected and valued, they can become more dedicated in turn. You may also want to give an employee referrer a bonus to show your appreciation.

- While in most cases an employee's motives should be "pure," there may be circumstances where a person wants to just work with their friend or receive the referral bonus. This can result in the candidate not being as qualified as either the referrer or referee said they were. The referrer may think that they can make up for the candidate's shortcomings or give them a crash course to level-set their skills. This can impact their own production in a negative way. And now your company may have two underperforming employees—and you may have to look to fill both of these positions in the not-so-far-off future.

11. CONCLUSION

We proposed a framework for job recommendation task. This framework facilitates the understanding 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

For this system to be hybrid, content-based filtering is required, which can only recommend jobs based on the user's current profile. It cannot deliver anything surprising based on the user's past searches. This paper also uses collaborative filtering which faces well-known problems of privacy breaches and cold start. The system has a broad scope that can be used to make it more robust and foolproof. Firstly, automating the crawling process is required, when a new company is added to the database. In other words, removing the one-time configuration step/process to fetch jobs of a particular new company can be done. These models can implement techniques such as KNN in collaborative filtering. Implementing NLP in content-based filtering for better and more accurate search matching can be done. Along with this, testing and collecting more user data for better performance of the collaborative filtering module is required. Lastly, improving the cleansing process of the job description and using natural language processing are required. While using

collaborative filtering, this work can be improved by giving different weights to different users based on their LinkedIn skills.

13. APPENDIX

13.1 SOURCE CODE

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>Job Recommender</title>
  <link rel="stylesheet" href="{ {url_for('static',filename='welcomestyle.css')}}">
  <script>
    window.watsonAssistantChatOptions = {
      integrationID: "65c01ed6-9fc1-4883-979a-3676279ebe44", // The ID of this integration.
      region: "us-south", // The region your integration is hosted in.
      serviceInstanceID: "8fcd017f-a192-420a-aafc-18cb0330efca", // 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>
      </ul>
    </nav>
  </header>
  <section id="sss1">
```

```

<center><h1 class="two" >Job Recommender</h1></center>
</section>
<br><br>
<div id="jobs">

    <center><h3 class="app">APPLY JOBS</h3></center><br>
    
    
    
    
    
    

    <h2 style="position:absolute;top: 450px;left:100px"><a
href="https://in.indeed.com/PHP-Developer-jobs?vjk=10cca9575b193c7d">APPLY</a></h2>
    <h2 style="position:absolute;top: 450px;left:700px"><a href="https://in.indeed.com/Software-
Developer-jobs?vjk=b7da08f07cac87d5">APPLY</a></h2>
    <h2 style="position:absolute;top: 450px;left:1200px"><a href="https://in.indeed.com/Web-
Developer-jobs?vjk=b81e49165da51eeb">APPLY</a></h2>
    <h2 style="position:absolute;top: 800px;left:1200px"><a href="https://in.indeed.com/SQL-
Developer-jobs?vjk=ac86b15908022123">APPLY</a></h2>
    <h2 style="position:absolute;top: 800px;left:700px"><a href="https://in.indeed.com/Java-
Developer-jobs?vjk=da306a665e00eb30">APPLY</a></h2>
    <h2 style="position:absolute;top: 800px;left:150px"><a href="https://in.indeed.com/Python-
Developer-jobs?vjk=fa7b9bd250044569">APPLY</a></h2>
</div>
</body>
</html>

```

REGISTER.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>Job Recommender</title>
<link rel="stylesheet" href="{ {url_for('static',filename='welcomestyle.css')}} ">
<script>
    window.watsonAssistantChatOptions = {
        integrationID: "65c01ed6-9fc1-4883-979a-3676279ebe44", // The ID of this integration.
        region: "us-south", // The region your integration is hosted in.
        serviceInstanceID: "8fcd017f-a192-420a-aafe-18cb0330efca", // 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>
        </ul>
    </nav>
</header>
<section id="sss1">
    <center><h1 class="two" >Job Recommender</h1></center>
</section>
<br><br>

```

```

<div id="jobs">

    <center><h3 class="app">APPLY JOBS</h3></center><br>

    <h2 style="position:absolute;top: 450px;left:100px"><a
href="https://in.indeed.com/PHP-Developer-jobs?vjk=10cca9575b193c7d">APPLY</a></h2>

    <h2 style="position:absolute;top: 450px;left:700px"><a href="https://in.indeed.com/Software-
Developer-jobs?vjk=b7da08f07cac87d5">APPLY</a></h2>

    <h2 style="position:absolute;top: 450px;left:1200px"><a href="https://in.indeed.com/Web-
Developer-jobs?vjk=b81e49165da51eeb">APPLY</a></h2>

    <h2 style="position:absolute;top: 800px;left:1200px"><a href="https://in.indeed.com/SQL-
Developer-jobs?vjk=ac86b15908022123">APPLY</a></h2>

    <h2 style="position:absolute;top: 800px;left:700px"><a href="https://in.indeed.com/Java-
Developer-jobs?vjk=da306a665e00eb30">APPLY</a></h2>

    <h2 style="position:absolute;top: 800px;left:150px"><a href="https://in.indeed.com/Python-
Developer-jobs?vjk=fa7b9bd250044569">APPLY</a></h2>

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

```

LOGIN.HTML

```

<html>

<head>

<title>Login page</title>

```

```
<link rel="stylesheet" href="{ { url_for('static',filename='registerstyle.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" >Job Recommender</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>
```

HOME.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>JOB RECOMMENDER</title>

  <link rel="stylesheet" href="{ {url_for('static',filename='homestyle.css')}}">

  <script>

    window.watsonAssistantChatOptions = {

      integrationID: "65c01ed6-9fc1-4883-979a-3676279ebe44", // The ID of this integration.

      region: "us-south", // The region your integration is hosted in.

      serviceInstanceID: "8fcd017f-a192-420a-aafe-18cb0330efca", // 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>

      </ul>

    </nav>

  </header>

</body>

</html>
```

```

        <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>

<hr>

<section id="sss1">

    <h1 class="two" >JOB RECOMMENDER</h1>

    <div id="ss1">

        <input class="input_search" type="text" placeholder="Search Jobs" >

        <button class="button_search">Search</button>

    </div>

</section>

<div>

</div>

</body>

</html>

```

FORGET.HTML

```

<html>

<head>

    <title>Login page</title>

    <link rel="stylesheet" href="{ {url_for('static',filename='registerstyle.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" >Job Recommender</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>

```


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>Job Recommender</title>
  <link rel="stylesheet" href="{ {url_for('static',filename='contactstyle.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" >Job Recommender</h1></center>
  </section>
  <br><br>
  <center><h1 class="h1lt">Contact us</h1></center>
  <form class="container">
    First name:<br>
    <input type="text" name="firstname" value="">
    <br>
    Last name:<br>
```

```
<input type="text" name="lastname" value="">
<br>
Email:<br>
<input type="text" name="email" value="">
<br>
<br>
<textarea name="message" rows="10" cols="30">
</textarea>
<br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

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>Job Recommender</title>
  <link rel="stylesheet" href="skillsstyle.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">Job Recommender</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>

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>SKYS.com</title>

<link rel="stylesheet" href="{ { url_for('static',filename='registerstyle.css') } } ">

</head>

<body>

<header>

<nav class="navbar">

Home

Register

Login

</nav>

</header>

<section id="sss1">

<center><h1 class="two" >SKYS.com</h1></center>

</section>

</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__)
app.secret_key='a'
try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SECURITY=SSL;SSL
ServerCertificat=DigiCertGlobalRootCA.crt;UID=chr84960;PWD=ky5p7N0JrFc3IbvX",",")
except:
    print("Unable to connect: ",ibm_db.conn_error())

@app.route("/")
def dash():
    return render_template('welcome.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(?,?,?,?,?)"

        prepare_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)
```

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
```

DOCKERFILE

```
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"]
```

SERVICE.YAML

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

13.2 GITHUB AND PROJECT DEMO LINK

GITHUB LINK: <https://github.com/IBM-EPBL/IBM-Project-41407-1660641811>

PROJECT DEMO LINK:

https://drive.google.com/file/d/1Ess7mtYrfNVbbDixVYb_36AHPkoX_DtX/view?usp=drivesdk