

Project Report

- 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
 - 6.3 Reports from JIRA
- 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**
 - 8.1 Test Cases
 - 8.2 User Acceptance Testing
- 9. RESULTS**
 - 9.1 Performance Metrics
- 10. ADVANTAGES & DISADVANTAGES**
- 11. CONCLUSION**
- 12. FUTURE SCOPE**
- 13. APPENDIX**
 - Source Code
 - GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project Overview

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.

1.2 Purpose

In the last few years, job recommender systems have become popular since they successfully reduce information overload by generating personalised job suggestions. Although in the literature there exists a variety of techniques and strategies. It is used as part of job recommender systems, most of them fail to recommending job vacancies that fit properly to the job seekers profiles

2. LITERATURE SURVEY

2.1 Existing problem

Existing works are mainly found for the company to select a candidate who is fit for their vacancy[17]. There are many experiments for calculating the four recommendation algorithm but with a different distance formula namely the Minkowski distance [5]. And some others are tried a different recommender system like collaborative which only helps when there are more data to relate. That won't help for a person who is searching that which job is the correct choice for him/her. R.J. Mooney and L. Roy used Content-Based Book Recommending[1] where the content-based recommendation helps for a cold start. And some authors also say that a content-based recommender is best when they researched a comparison study of job recommendations [9]. A recommender system is not only the main part of accurate prediction. There are some other things like vectorizing the words and then similarity functions. Authors like Shouning Qu[3], and Li-Ping Jing[2] said that for text mining, tf-idf is the best approach for text feature selection. Ravali Boorugu has researched NLP and tried various text summarization techniques[14][19]. Some papers also say about similarity detection with many languages [8][10]. Jeevamol Joy and Renumol V G [16] discussed which similarity is the best one for a content-based recommendation system. They finally concluded that cosine similarity is the best similarity for content based recommender systems. Cosine similarity is not only used for recommender systems but is used to find the similarity functions between two sentences or two paragraphs[8][20]. Mohammad Alobed has tried "A Comparative Analysis of Euclidean, Jaccard, and Cosine Similarity Measure and Arabic Wordnet for Automated Arabic Essay Scoring"[21], and L. Zahrotum also compared jaccard, Jaccardidean and cosine similarity. They both said that Cosine similarity with all stemming types has the lowest error compared with the Jaccard and Euclidean similarity[7]. There is already a system that works with both tf-idf and cosine similarity recommendations. It is used for patient support forums[6]. Tanya V. Yadalam, Vaishnavi, M. Gowda, and Vanditha researched those career recommendations content-based filtering which was mostly like my project but inside it, they mostly discussed security, transparency for the data, and the framework [15]. Most works are just built for the companies or for the purpose of making money from the people by giving some irrelevant choices. Many were using collaborative recommendation, which recommends the many searched jobs or the jobs which were chosen by some other. It only works if the system deals with a larger number of resumes which seems it can only be used by the companies. Some systems are asking to log in and some are asked to buy subscriptions. Logging in makes you redirect some spam mails. In many papers, they have been solved through a content recommendation which is not enough. A literature paper[15] had done research on content recommender system, vectorizer, and cosine similarity in a row but in that the author doesn't think about the implementation process and only concentrated more on securing the data

2.2 References

- <https://ijarcce.com/wp-content/uploads/2021/09/IJARCCCE.2021.10809.pdf>
- <https://www.irjet.net/archives/V7/i9/IRJET-V7I9633.pdf>
- <https://scholarspace.manoa.hawaii.edu/handle/10125/50379>
- [Job Seekers' Acceptance of Job Recommender Systems: Results of an Empirical Study (hawaii.edu)]
- <https://ieeexplore.ieee.org/document/7944917>
- https://www.researchgate.net/publication/304413319_FoDRA_-_A_new_content-based_job_recommendation_algorithm_for_job_seeking_and_recruiting

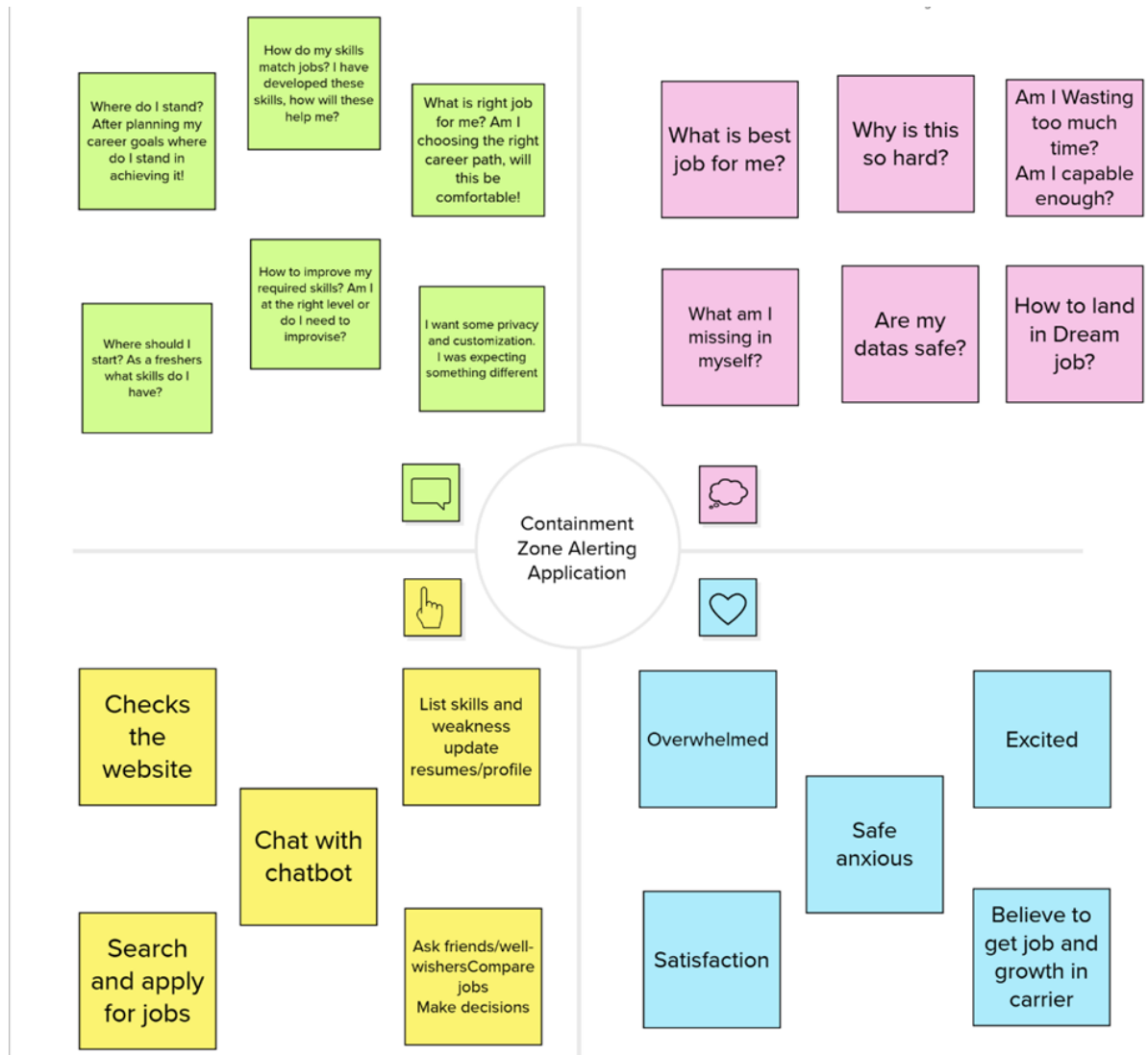
<https://www.semanticscholar.org/paper/Job-Recommender-Systems%3A-A-Review-Ruijt-Bhulai/39d3e2b5fd5cc01032b7c74b431f5cc348f8c8ef>
<https://arxiv.org/abs/1907.12379>
<https://core.ac.uk/download/pdf/228066334.pdf>

2.3 Problem Statement Definition

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.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

We are providing job opportunities for job seekers and providing additional information about jobs to seek their career brighter. Candidates have an opportunity to view the company's information. They get a list of all job opportunities and check photos.

3.3 Proposed Solution

We are proposing an application which will help the job seekers to give suggestions on jobs based on their skills. The most effective recommendation shows a strong connection between the candidate's skills and experience and those required for success in the position. They are applying for the job listing and the person's resume, and think of the ways in which the person has demonstrated the skills necessary for the job.

3.4 Problem Solution fit

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. Registration through Application.
FR-2	User Confirmation	Confirmation via Email. Confirmation via OTP.
FR-3	User Login	Login using credentials.
FR-4	User Application	Search for desired company.
FR-5	User Profile	Complete user profile by providing personal details.
FR-6	User Application	User applies for the desired company.

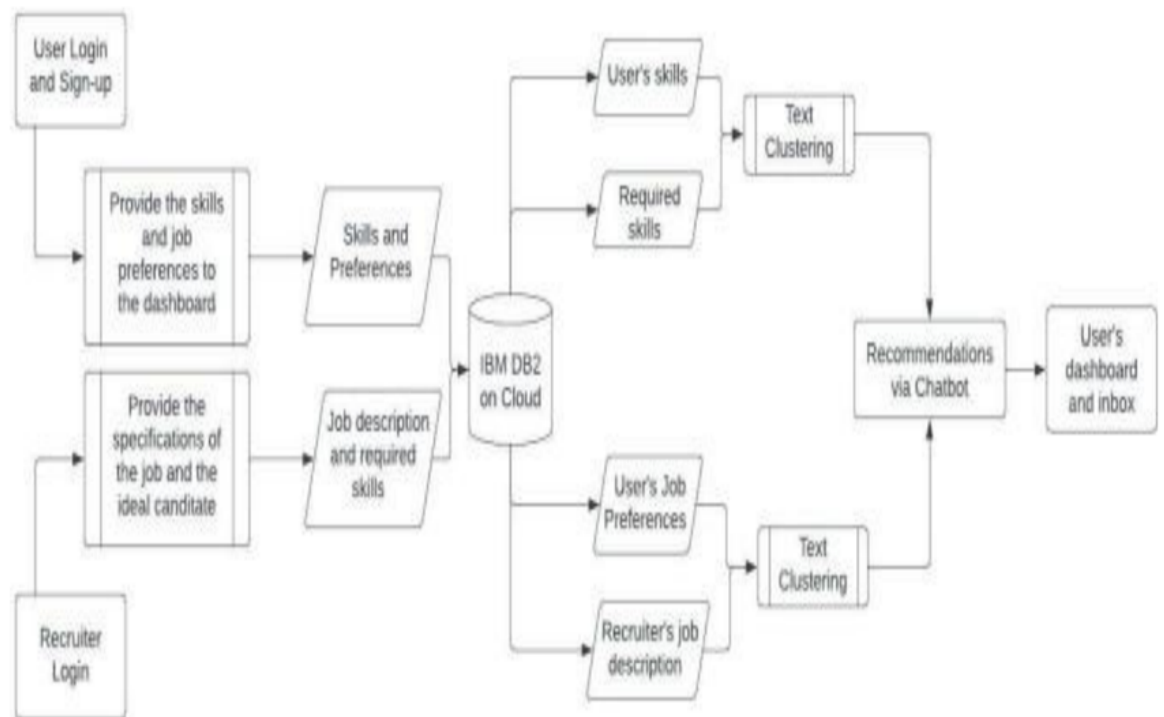
4.2 Non-Functional requirements

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	✓ User-Friendly Application.
NFR-2	Security	✓ End-to-End Encryption.
NFR-3	Reliability	✓ Based on personalised skill sets.

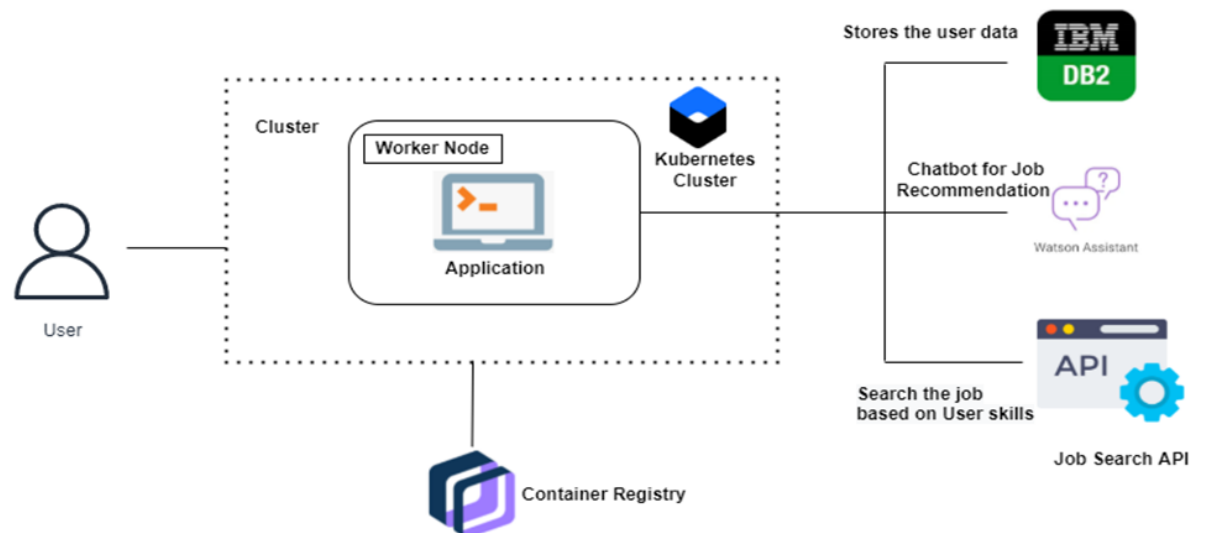
NFR-4	Performance	✓ Analysing the skill sets of the user to ensure our recommendations reach them better.
NFR-5	Availability	✓ 24/7 chatbot support✓ 24/7 chatbot support.
NFR-6	Scalability	✓ Reaching the on-scale requirement of the user.

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

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 and Login Page	10	Medium	Tharun Raj, Deepak and Gunabalan
Sprint-1	Database Connectivity	USN-2	Viewing and applying jobs Connecting UI with Database	10	High	Sakthivel and Gunabalan
Sprint-2	SendGrid Integration	USN-3	SendGrid Integration with Python Code	10	Low	Deepak and Gunabalan
Sprint-2	Chatbot Development	USN-4	Building a chatbot	10	High	Deepak and Sakthivel
Sprint-3	Integration and Containerization	USN-5	Integrating chatbot to the HTML page and containerizing the application	20	Medium	Sakthivel and Tharun Raj
Sprint-4	Upload Image and Deployment	USN-6	Upload the image to the IBM Registry and deploy it in the Kubernetes Cluster.	20	High	Deepak and Sakthivel

6.2 Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date	Story Points Completed
Sprint 1	20	5 Days	24 Oct 2022	28 Oct 2022	20
Sprint 2	20	7 Days	31 Oct 2022	06 Nov 2022	20
Sprint 3	20	7 Days	07 Nov 2022	13 Nov 2022	20

6.3 Reports from JIRA

Average Velocity for a 10 day sprint which earns us 20 points is 2.

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

7.1 Feature 1

HTML code

Sign Up Page

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

{% block head %}
    <title>sign up - Job Hunter</title>
    <link rel="stylesheet" href="static/css/bootstrap.min.css">
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway:300italic,400italic,600italic,700italic,800italic,400,300,600,700,800&display=swap">
{% endblock %}

{% block content %}

    <nav class="navbar navbar-light navbar-expand-md fixed-top navbar-shrink py-3" id="mainNav">
        <div class="container"><a class="navbar-brand d-flex align-items-center" href="/"></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
        <a href="/" ></a>
        <div class="collapse navbar-collapse" id="navcol-1">
            <ul class="navbar-nav mx-auto">
                <li class="nav-item"><a class="nav-link" href="/">Home</a></li>
                <li class="nav-item"></li>
                <li class="nav-item"></li>
                <li class="nav-item"><a class="nav-link" href="discover">Discover</a></li>
                <li class="nav-item"><a class="nav-link" href="contacts">Contacts</a></li>
                <li class="nav-item"><a class="nav-link" href="login">Log in</a></li>
            </ul><a class="btn btn-primary shadow" role="button" href="signup">Sign up</a>
        </div>
    </nav>

{% endblock %}
```

```

    </div>
  </div>
</nav>
<section class="py-4 py-md-5 my-5">
  <div class="container py-md-5">
    <div class="row">
      <div class="col-md-6 text-center"></div>
      <div class="col-md-5 col-xl-4 text-center
text-md-start">
        <h2 class="display-6 fw-bold mb-5"><span
class="underline pb-1"><strong>Sign up</strong></span></h2>
        <form method="post">
          <div class="mb-3"><input class="shadow
form-control" type="text" name="name" placeholder="Name"></div>
          <div class="mb-3"><input class="shadow
form-control" type="number" name="phone" placeholder="Phone
number"></div>
          <div class="mb-3"><input class="shadow
form-control" type="email" name="email"
placeholder="Email"></div>
          <div class="mb-3"><input class="shadow
form-control" type="password" name="password"
placeholder="Password"></div>
          <div class="mb-5"><button class="btn
btn-primary shadow" type="submit">Create account</button></div>
          <p class="text-muted">Already have an
account? <a href="login">Log in</a><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
viewBox="0 0 24 24" stroke-width="2" stroke="currentColor"
fill="none" stroke-linecap="round" stroke-linejoin="round"
class="icon icon-tabler icon-tabler-arrow-narrow-right">
            <path stroke="none" d="M0
0h24v24H0z" fill="none"></path>
            <line x1="5" y1="12" x2="19"
y2="12"></line>
            <line x1="15" y1="16" x2="19"
y2="12"></line>
            <line x1="15" y1="8" x2="19"
y2="12"></line>
          </svg></a></p>
        <div class="mb-3"></div>
      </div>
    </div>
  </div>

```

```

        <div class="mb-3"></div>
        <div></div>
    </form>
</div>
</div>
</div>
</section>
<footer>
    <div class="container py-4 py-lg-5">
        <div class="row row-cols-2 row-cols-md-4">
            <div class="col-12 col-md-3">
                <div class="fw-bold d-flex align-items-center mb-2"><span>Job Hunter</span></div>
                <p class="text-muted"><span style="color: rgb(32, 33, 36);">The most powerful platform for job seekers as well as recruiter</span><br><br><br></p>
            </div>
            <div class="col-sm-4 col-md-3 text-lg-start d-flex flex-column">
                <h3 class="fs-6 fw-bold">Discover</h3>
                <ul class="list-unstyled">
                    <li><a href="#">Web design</a></li>
                    <li><a href="#">Development</a></li>
                    <li><a href="#">Hosting</a></li>
                </ul>
            </div>
            <div class="col-sm-4 col-md-3 text-lg-start d-flex flex-column">
                <h3 class="fs-6 fw-bold">About</h3>
                <ul class="list-unstyled">
                    <li><a href="#">Company</a></li>
                    <li><a href="#">Team</a></li>
                    <li><a href="#">Legacy</a></li>
                </ul>
            </div>
            <div class="col-sm-4 col-md-3 text-lg-start d-flex flex-column">
                <h3 class="fs-6 fw-bold">Careers</h3>
                <ul class="list-unstyled">
                    <li><a href="#">Job openings</a></li>
                    <li><a href="#">Employee success</a></li>
                    <li><a href="#">Benefits</a></li>
                </ul>
            </div>
        </div>
    </div>

```

```
</div>
</div>
<hr>
<div class="text-muted d-flex justify-content-between align-items-center pt-3">
    <p class="mb-0"><strong>Copyright © 2022 Job Hunter</strong>&nbsp;</p>
    <p class="mb-0"><strong>Contributers:</strong>
    <br>
    Deepak R,Gunabalan P,Tharun Raj S,Sakthivel M
    &nbsp;</p>
    <ul class="list-inline mb-0">
        <li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-facebook">
            <path d="M16
8.049c0-4.446-3.582-8.05-8-8.05C3.58 0-.002 3.603-.002 8.05c0
4.017 2.926 7.347 6.75 7.951v-5.625h-2.03V8.05H6.75V6.275c0-2.017
1.195-3.131 3.022-3.131.876 0 1.791.157
1.791.157v1.98h-1.009c-.993 0-1.303.621-1.303
1.258v1.51h2.218l-.354 2.326H9.25V16c3.824-.604 6.75-3.934
6.75-7.951z"></path>
        </svg></li>
        <li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-twitter">
            <path d="M5.026 15c6.038 0
9.341-5.003 9.341-9.334 0-.14 0-.282-.006-.422A6.685 6.685 0 0 0
16 3.542a6.658 6.658 0 0 1-1.889.518 3.301 3.301 0 0 0 0 0
1.447-1.817 6.533 6.533 0 0 1-2.087.793A3.286 3.286 0 0 0 7.875
6.03a9.325 9.325 0 0 1-6.767-3.429 3.289 3.289 0 0 0 1.018
4.382A3.323 3.323 0 0 1 .64 6.575v.045a3.288 3.288 0 0 0 2.632
3.218 3.203 3.203 0 0 1-.865.115 3.23 3.23 0 0 1-.614-.057 3.283
3.283 0 0 0 3.067 2.277A6.588 6.588 0 0 1 .78 13.58a6.32 6.32 0 0
1-.78-.045A9.344 9.344 0 0 0 5.026 15z"></path>
        </svg></li>
        <li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-instagram">
            <path d="M8 0C5.829 0 5.556.01
4.703.048 3.85.088 3.269.222 2.76.42a3.917 3.917 0 0 0
0-1.417.923A3.927 3.927 0 0 0 .42 2.76C.222 3.268.087 3.85.048
4.7.01 5.555 0 5.827 0 8.001c0 2.172.01 2.444.048
```

```

3.297.04.852.174      1.433.372      1.942.205.526.478.972.923
1.417.444.445.89.719  1.416.923.51.198  1.09.333  1.942.372C5.555
15.99  5.827  16  8  16s2.444-.01  3.298-.048c.851-.04  1.434-.174
1.943-.372a3.916      3.916      0      0      0
1.416-.923c.445-.445.718-.891.923-1.417.197-.509.332-1.09.372-1.9
42C15.99      10.445      16      10.173      16
8s-.01-2.445-.048-3.299c-.04-.851-.175-1.433-.372-1.941a3.926
3.926      0      0      0-.923-1.417A3.911      3.911      0      0      0
13.24.42c-.51-.198-1.092-.333-1.943-.372C10.443.01  10.172  0  7.998
0h.003zm-.717  1.442h.718c2.136  0  2.389.007  3.232.046.78.035
1.204.166
1.486.275.373.145.64.319.92.599.28.28.453.546.598.92.11.281.24.70
5.275      1.485.039.843.047      1.096.047      3.231s-.008      2.389-.047
3.232c-.035.78-.166      1.203-.275      1.485a2.47      2.47      0      0
1-.599.919c-.28.28-.546.453-.92.598-.28.11-.704.24-1.485.276-.843
.038-1.096.047-3.232.047s-2.39-.009-3.233-.047c-.78-.036-1.203-.1
66-1.485-.276a2.478  2.478  0  0  1-.92-.598  2.48  2.48  0  0
1-.6-.92c-.109-.281-.24-.705-.275-1.485-.038-.843-.046-1.096-.046
-3.233
0-2.136.008-2.388.046-3.231.036-.78.166-1.204.276-1.486.145-.373.
319-.64.599-.92.28-.28.546-.453.92-.598.282-.11.705-.24
1.485-.276.738-.034      1.024-.044      2.515-.045v.002zm4.988
1.328a.96.96  0  1  0  0  1.92.96.96  0  0  0  0-1.92zm-4.27  1.122a4.109
4.109  0  1  0  0  8.217  4.109  4.109  0  0  0  0-8.217zm0  1.441a2.667
2.667  0  1  1  0  5.334  2.667  2.667  0  0  1  0-5.334z"></path>
      </svg></li>
    </ul>
  </div>
</div>
</footer>
<script src="static/js/bootstrap.min.js"></script>
<script src="static/js/script.min.js"></script>

{% endblock %}

```

HTML code

Login Page

```

{% extends 'base.html' %}

{% block head %}
  <title>Log in - Job Hunter</title>

```

```

<link rel="stylesheet" href="static/css/bootstrap.min.css">
<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway:300italic,400italic,600italic,700italic,800italic,400,300,600,700,800&display=swap">
{% endblock %}

{% block content %}

<nav class="navbar navbar-light navbar-expand-md fixed-top navbar-shrink py-3" id="mainNav">
    <div class="container"><a class="navbar-brand d-flex align-items-center" href="/"></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
        <a href="/" ></a>
        <div class="collapse navbar-collapse" id="navcol-1">
            <ul class="navbar-nav mx-auto">
                <li class="nav-item"><a class="nav-link" href="/">Home</a></li>
                <li class="nav-item"></li>
                <li class="nav-item"></li>
                <li class="nav-item"><a class="nav-link" href="discover">Discover</a></li>
                <li class="nav-item"><a class="nav-link" href="contacts">Contacts</a></li>
                <li class="nav-item"><a class="nav-link active" href="login">Log in</a></li>
            </ul><a class="btn btn-primary shadow" role="button" href="signup">Sign up</a>
        </div>
    </div>
</nav>

<section class="py-4 py-md-5 my-5">
    <div class="container py-md-5">
        <div class="row">
            <div class="col-md-6 text-center"></div>

```



```

<div class="col-md-5 col-xl-4 text-center
text-md-start">
    <h2 class="display-6 fw-bold mb-5"><span
class="underline pb-1"><strong>Login</strong><br></span></h2>
    <form method="post">
        <div class="mb-3"><input class="shadow
form-control"
type="email"
name="email"
placeholder="Email"></div>
        <div class="mb-3"><input class="shadow
form-control"
type="password"
name="password"
placeholder="Password"></div>
        <div class="mb-5"><button class="btn
btn-primary shadow" type="submit">Log in</button></div>
        <p class="text-muted"><a
href="forgot">Forgot your password?</a></p>
    </form>
</div>
</div>
</div>
</section>
<footer>
    <div class="container py-4 py-lg-5">
        <div class="row row-cols-2 row-cols-md-4">
            <div class="col-12 col-md-3">
                <div class="fw-bold d-flex align-items-center
mb-2"><span>Job Hunter</span></div>
                <p class="text-muted"><span style="color:
rgb(32, 33, 36);">The most powerful platform for job seekers as
well as recruiter</span><br><br><br><br></p>
            </div>
            <div class="col-sm-4 col-md-3 text-lg-start
d-flex flex-column">
                <h3 class="fs-6 fw-bold">Discover</h3>
                <ul class="list-unstyled">
                    <li><a href="#">Web design</a></li>
                    <li><a href="#">Development</a></li>
                    <li><a href="#">Hosting</a></li>
                </ul>
            </div>
            <div class="col-sm-4 col-md-3 text-lg-start
d-flex flex-column">
                <h3 class="fs-6 fw-bold">About</h3>
                <ul class="list-unstyled">

```

```

        <li><a href="#">Company</a></li>
        <li><a href="#">Team</a></li>
        <li><a href="#">Legacy</a></li>
    </ul>
</div>
</div>
<div class="col-sm-4 col-md-3 text-lg-start
d-flex flex-column">
    <h3 class="fs-6 fw-bold">Careers</h3>
    <ul class="list-unstyled">
        <li><a href="#">Job openings</a></li>
        <li><a href="#">Employee success</a></li>
        <li><a href="#">Benefits</a></li>
    </ul>
</div>
</div>
<hr>
<div class="text-muted d-flex justify-content-between
align-items-center pt-3">
    <p class="mb-0"><strong>Copyright © 2022 Job
Hunter</strong>&nbsp;</p>
    <p class="mb-0"><strong>Contributors:</strong>
    <br>
    Deepak R,Gunabalan P,Tharun Raj S,Sakthivel M
    &nbsp;</p>
    <ul class="list-inline mb-0">
        <li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-facebook">
            <path d="M16
8.049c0-4.446-3.582-8.05-8.05-8.05C3.58 0-.002 3.603-.002 8.05c0
4.017 2.926 7.347 6.75 7.951v-5.625h-2.03V8.05H6.75V6.275c0-2.017
1.195-3.131 3.022-3.131.876 0 1.791.157
1.791.157v1.98h-1.009c-.993 0-1.303.621-1.303
1.258v1.51h2.218l-.354 2.326H9.25V16c3.824-.604 6.75-3.934
6.75-7.951z"></path>
        </svg></li>
        <li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-twitter">
            <path d="M5.026 15c6.038 0
9.341-5.003 9.341-9.334 0-.14 0-.282-.006-.422A6.685 6.685 0 0 0
16 3.542a6.658 6.658 0 0 0 1-1.889.518 3.301 3.301 0 0 0
1.447-1.817 6.533 6.533 0 0 1-2.087.793A3.286 3.286 0 0 0 7.875

```

```
6.03a9.325 9.325 0 0 1-6.767-3.429 3.289 3.289 0 0 0 1.018
4.382A3.323 3.323 0 0 1 .64 6.575v.045a3.288 3.288 0 0 0 2.632
3.218 3.203 3.203 0 0 1-.865.115 3.23 3.23 0 0 1-.614-.057 3.283
3.283 0 0 0 3.067 2.277A6.588 6.588 0 0 1 .78 13.58a6.32 6.32 0 0
1-.78-.045A9.344 9.344 0 0 0 5.026 15z"></path>
</svg></li>
<li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-instagram">
<path d="M8 0C5.829 0 5.556.01
4.703.048 3.85.088 3.269.222 2.76.42a3.917 3.917 0 0
0-1.417.923A3.927 3.927 0 0 0 .42 2.76C.222 3.268.087 3.85.048
4.7.01 5.555 0 5.827 0 8.001c0 2.172.01 2.444.048
3.297.04.852.174 1.433.372 1.942.205.526.478.972.923
1.417.444.445.89.719 1.416.923.51.198 1.09.333 1.942.372C5.555
15.99 5.827 16 8 16s2.444-.01 3.298-.048c.851-.04 1.434-.174
1.943-.372a3.916 3.916 0 0 0 0 0
1.416-.923c.445-.445.718-.891.923-1.417.197-.509.332-1.09.372-1.9
42C15.99 10.445 16 10.173 16
8s-.01-2.445-.048-3.299c-.04-.851-.175-1.433-.372-1.941a3.926
3.926 0 0 0-.923-1.417A3.911 3.911 0 0 0 0
13.24.42c-.51-.198-1.092-.333-1.943-.372C10.443.01 10.172 0 7.998
0h.003zm-.717 1.442h.718c2.136 0 2.389.007 3.232.046.78.035
1.204.166
1.486.275.373.145.64.319.92.599.28.28.453.546.598.92.11.281.24.70
5.275 1.485.039.843.047 1.096.047 3.231s-.008 2.389-.047
3.232c-.035.78-.166 1.203-.275 1.485a2.47 2.47 0 0
1-.599.919c-.28.28-.546.453-.92.598-.28.11-.704.24-1.485.276-.843
.038-1.096.047-3.232.047s-2.39-.009-3.233-.047c-.78-.036-1.203-.1
66-1.485-.276a2.478 2.478 0 0 1-.92-.598 2.48 2.48 0 0
1-.6-.92c-.109-.281-.24-.705-.275-1.485-.038-.843-.046-1.096-.046
-3.233
0-2.136.008-2.388.046-3.231.036-.78.166-1.204.276-1.486.145-.373.
319-.64.599-.92.28-.28.546-.453.92-.598.282-.11.705-.24
1.485-.276.738-.034 1.024-.044 2.515-.045v.002zm4.988
1.328a.96.96 0 1 0 0 1.92.96.96 0 0 0 0-1.92zm-4.27 1.122a4.109
4.109 0 1 0 0 8.217 4.109 4.109 0 0 0 0-8.217zm0 1.441a2.667
2.667 0 1 1 0 5.334 2.667 2.667 0 0 1 0-5.334z"></path>
</svg></li>
</ul>
</div>
</div>
</footer>
```

```

<script src="static/js/bootstrap.min.js"></script>
<script src="static/js/script.min.js"></script>

{% endblock %}

```

Python code

When a user Signs up the credentials should be added to database, when a user logs in the credentials need to be verified

```

@app.route('/signup', methods=['POST', 'GET'])
def signup():
    if request.method == 'POST':
        # conn = connection()
        try:
            sql = "INSERT INTO USERS1
VALUES ('{}','{}','{}','{}').format(request.form["name"],request.
form["phone"],request.form["email"],request.form["password"])
            print(sql)
            ibm_db.exec_immediate(conn,sql)
            print('ss')
            return render_template('login.html')
        except:
            return render_template('signup.html')
    else:
        return render_template('signup.html')

@app.route('/login', methods=['POST', 'GET'])
def login():
    if request.method == 'POST':
        email = request.form["email"]
        password = request.form["password"]
        sql = "SELECT COUNT(*) FROM USERS1 WHERE EMAIL=? AND
PASSWORD=?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt, 1, email)
        ibm_db.bind_param(stmt, 2, password)
        ibm_db.execute(stmt)
        res = ibm_db.fetch_assoc(stmt)
        if res['1'] == 1:
            session['loggedin'] = True
            session['email'] = email
            return render_template('userpage.html')
        else:
            return render_template('login.html')

```

```

else:
    return render_template('login.html')

```

7.2 Feature 2

HTML code

Discover page which shows users the available jobs that they can apply to

```

{% extends 'base.html' %}

{% block head %}
    <title>Discover - Job Hunter</title>
    <link rel="stylesheet" href="static/css/bootstrap.min.css">
    <link rel="stylesheet" href="static/css/style.css">
    <link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Raleway:300italic,400italic,600italic,700italic,800italic,400,300,600,700,800&display=swap">
{% endblock %}

{% block content %}

    <nav class="navbar navbar-light navbar-expand-md fixed-top navbar-shrink py-3" id="mainNav">
        <div class="container"><a class="navbar-brand d-flex align-items-center" href="/"></a><button data-bs-toggle="collapse" class="navbar-toggler" data-bs-target="#navcol-1"><span class="visually-hidden">Toggle navigation</span><span class="navbar-toggler-icon"></span></button>
        <a href="/" ></a>
        <div class="collapse navbar-collapse" id="navcol-1">
            <ul class="navbar-nav mx-auto">
                <li class="nav-item"><a class="nav-link" href="/">Home</a></li>
                <li class="nav-item"><a class="nav-link active" href="#">Discover</a></li>
                <li class="nav-item"><a class="nav-link" href="contacts">Contacts</a></li>
                <li class="nav-item"><a class="nav-link" href="login">Log in</a></li>
            </ul>
        </div>
    </nav>

```

```

        </ul><a class="btn btn-primary shadow"
role="button" href="signup">Sign up</a>
        </div>
    </div>
</nav>
<div class="container-xxl py-5">
    <div class="container">
        <h1 class="text-center mb-5 wow fadeInUp"
data-wow-delay="0.1s">Job Listing</h1>
        <div class="tab-class text-center wow fadeInUp"
data-wow-delay="0.3s">
            <div class="tab-content">
                <div id="tab-1" class="tab-pane fade show p-0
active">
                    <div class="job-item p-4 mb-4">
                        {% for job in res %}
                        <div class="row g-4">
                            <div class="col-sm-12 col-md-8
d-flex align-items-center">
                                <div class="text-start ps-4">
                                    <h5
class="mb-3">{{ job["NAME"] }}</h5>
                                    <span
class="text-truncate me-3"><i class="fa fa-map-marker-alt
text-primary me-2"></i>{{ job["LOCATION"] }}</span>
                                    <span
class="text-truncate me-3"><i class="far fa-clock text-primary
me-2"></i>{{ job["TYPE"] }}</span>
                                    <span
class="text-truncate me-0"><i class="far fa-money-bill-alt
text-primary me-2"></i>{{ job["SALARY"] }}</span>
                                </div>
                            </div>
                            <div class="col-sm-12 col-md-4
d-flex flex-column align-items-start align-items-md-end
justify-content-center">
                                <div class="d-flex mb-3">
                                    <a class="btn btn-light
btn-square me-3" href=""><i class="far fa-heart
text-primary"></i></a>
                                    <a class="btn
btn-primary" href="">Apply Now</a>
                                </div>

```

```

<small
class="text-truncate"><i class="far fa-calendar-alt text-primary
me-2"></i>DEAD LINE: {{job["DEADLINE"]}}</small>

    </div>

    </div>

    {% endfor %}

</div>

    <a class="btn btn-primary py-3 px-5"
href="">Browse More Jobs</a>

    </div>

</div>

</div>

</div>

</div>

</div>

<div class="container py-4 py-lg-5">
    <div class="row row-cols-2 row-cols-md-4">
        <div class="col-12 col-md-3">
            <div class="fw-bold d-flex align-items-center
mb-2"><span>Job Hunter</span></div>
            <p class="text-muted"><span style="color:
rgb(32, 33, 36);">The most powerful platform for job seekers as
well as recruiter</span><br><br><br><br></p>
        </div>
        <div class="col-sm-4 col-md-3 text-lg-start
d-flex flex-column">
            <h3 class="fs-6 fw-bold">Discover</h3>
            <ul class="list-unstyled">
                <li><a href="#">Web design</a></li>
                <li><a href="#">Development</a></li>
                <li><a href="#">Hosting</a></li>
            </ul>
        </div>
        <div class="col-sm-4 col-md-3 text-lg-start
d-flex flex-column">
            <h3 class="fs-6 fw-bold">About</h3>
            <ul class="list-unstyled">
                <li><a href="#">Company</a></li>
                <li><a href="#">Team</a></li>
                <li><a href="#">Legacy</a></li>
            </ul>
        </div>
    </div>

```



```

        </svg></li>
        <li class="list-inline-item"><svg
xmlns="http://www.w3.org/2000/svg" width="1em" height="1em"
fill="currentColor" viewBox="0 0 16 16" class="bi bi-instagram">
        <path d="M8 0C5.829 0 5.556.01
4.703.048 3.85.088 3.269.222 2.76.42a3.917 3.917 0 0
0-1.417.923A3.927 3.927 0 0 0 .42 2.76C.222 3.268.087 3.85.048
4.7.01 5.555 0 5.827 0 8.001c0 2.172.01 2.444.048
3.297.04.852.174 1.433.372 1.942.205.526.478.972.923
1.417.444.445.89.719 1.416.923.51.198 1.09.333 1.942.372C5.555
15.99 5.827 16 8 16s2.444-.01 3.298-.048c.851-.04 1.434-.174
1.943-.372a3.916 3.916 0 0 0 0 0 0 0
1.416-.923c.445-.445.718-.891.923-1.417.197-.509.332-1.09.372-1.9
42C15.99 10.445 16 10.173 16
8s-.01-2.445-.048-3.299c-.04-.851-.175-1.433-.372-1.941a3.926
3.926 0 0 0-.923-1.417A3.911 3.911 0 0 0
13.24.42c-.51-.198-1.092-.333-1.943-.372C10.443.01 10.172 0 7.998
0h.003zm-.717 1.442h.718c2.136 0 2.389.007 3.232.046.78.035
1.204.166
1.486.275.373.145.64.319.92.599.28.28.453.546.598.92.11.281.24.70
5.275 1.485.039.843.047 1.096.047 3.231s-.008 2.389-.047
3.232c-.035.78-.166 1.203-.275 1.485a2.47 2.47 0 0
1-.599.919c-.28.28-.546.453-.92.598-.28.11-.704.24-1.485.276-.843
.038-1.096.047-3.232.047s-2.39-.009-3.233-.047c-.78-.036-1.203-.1
66-1.485-.276a2.478 2.478 0 0 1-.92-.598 2.48 2.48 0 0
1-.6-.92c-.109-.281-.24-.705-.275-1.485-.038-.843-.046-1.096-.046
-3.233
0-2.136.008-2.388.046-3.231.036-.78.166-1.204.276-1.486.145-.373.
319-.64.599-.92.28-.28.546-.453.92-.598.282-.11.705-.24
1.485-.276.738-.034 1.024-.044 2.515-.045v.002zm4.988
1.328a.96.96 0 1 0 0 1.92.96.96 0 0 0 0-1.92zm-4.27 1.122a4.109
4.109 0 1 0 0 8.217 4.109 4.109 0 0 0 0-8.217zm0 1.441a2.667
2.667 0 1 1 0 5.334 2.667 2.667 0 0 1 0-5.334z"></path>
        </svg></li>
    </ul>
</div>
</div>
</footer>
<script
src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0/dist/js/bootstr
ap.bundle.min.js"></script>

```

```
<script src="static/js/bootstrap.min.js"></script>
<script src="static/js/script.min.js"></script>

{% endblock %}
```

Python code

To retrieve data from the database to show the user the available jobs

```
@app.route('/discover')
def discover():
    stmt=ibm_db.exec_immediate(conn, "SELECT * FROM JOBS")
    lt = []
    while ibm_db.fetch_row(stmt) != False:
        lt.append({"NAME":ibm_db.result(stmt,
0),"LOCATION":ibm_db.result(stmt, 1),"TYPE":ibm_db.result(stmt,
2),"SALARY":ibm_db.result(stmt, 3),"DEADLINE":ibm_db.result(stmt,
4)})

    return render_template('discover.html', res=lt)
```

7.3 Database Schema (if Applicable)

Database for the jobs displayed

JOBS

No statistics available.

Name	Data type	Nullable	Length	Scale	
NAME	CHAR	N	50	0	🔍
LOCATION	CHAR	N	50	0	🔍
TYPE	CHAR	N	50	0	🔍
SALARY	CHAR	N	50	0	🔍
DEADLINE	CHAR	N	50	0	🔍

Database for the signup and login

USERS1

No statistics available.

Name	Data type	Nullable	Length	Scale	
NAME	CHAR	Y	20	0	🙁
PHONE	CHAR	Y	20	0	🙁
EMAIL	CHAR	Y	20	0	🙁
PASSWORD	CHAR	Y	20	0	🙁

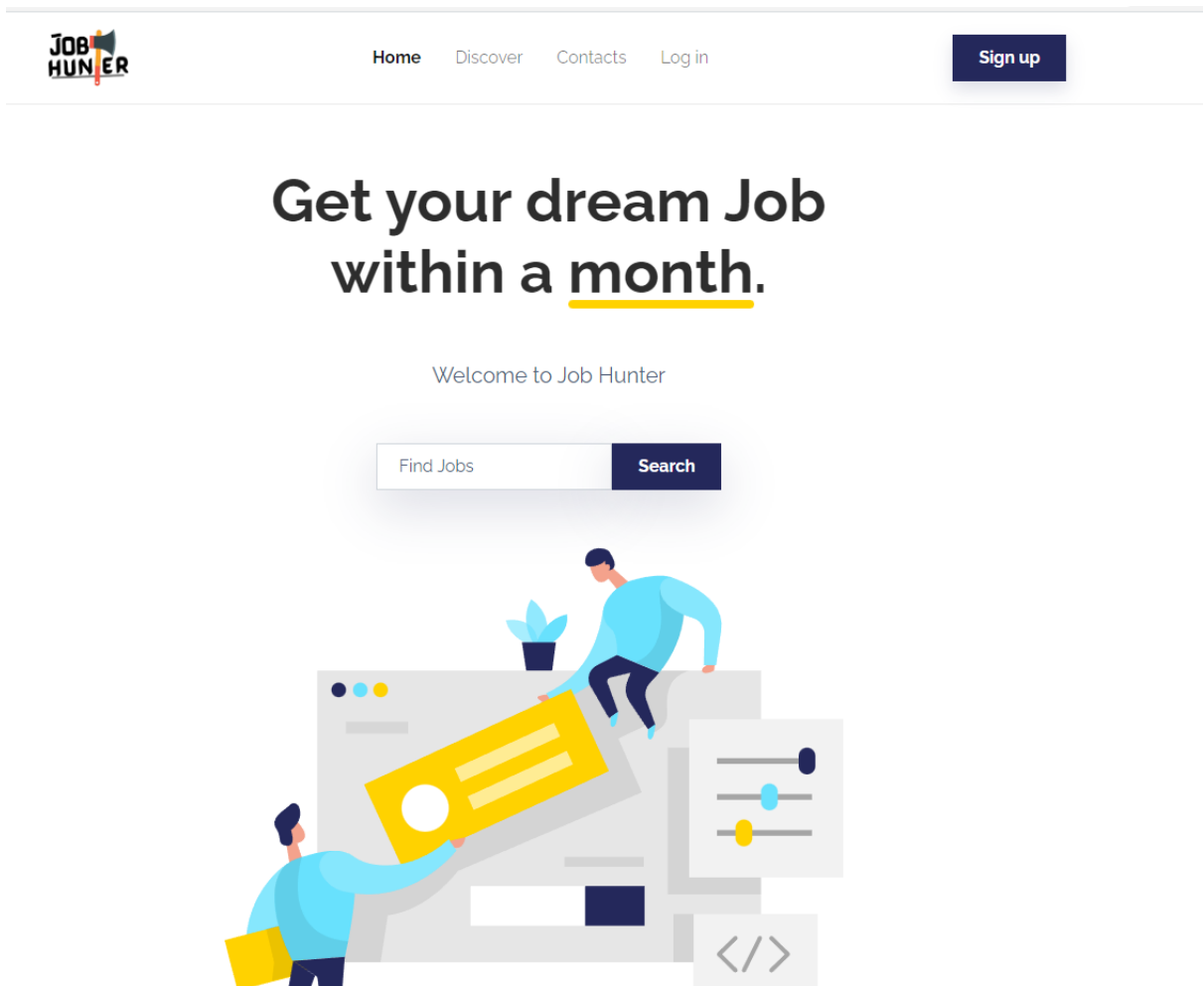
8. TESTING

8.1 Test Cases

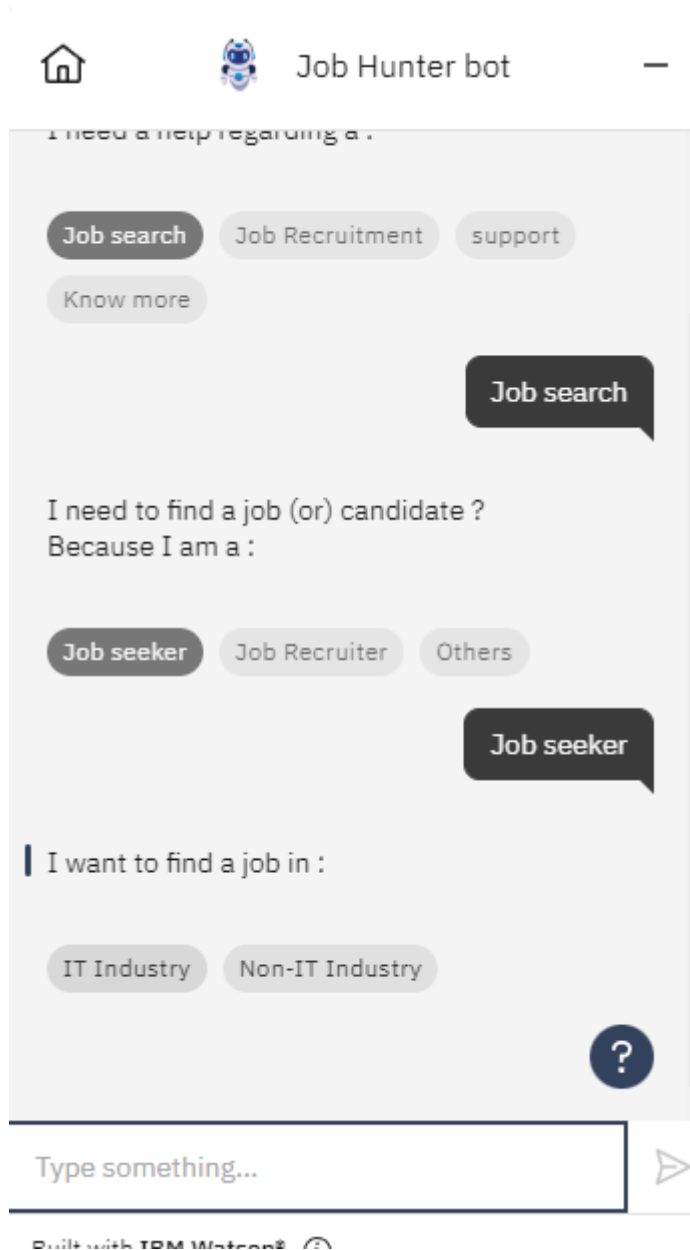
8.2 User Acceptance Testing

9. RESULTS

Front Page - Home Page



Chatbot



View Jobs

[HOME](#)[DISCOVER](#)[CONTACTS](#)[LOG IN](#)[Sign up](#)

SOFTWARE ENGINEER

NOIDA, INDIA FULL TIME 8LPA

[Apply Now](#)

DEAD LINE: 25-11-2022

UI Engineer

Gurugram, India Part Time 5lpa

[Apply Now](#)

DEAD LINE: 05-12-2022

UI Engineer

Gurugram, India Part Time 5lpa

[Apply Now](#)

DEAD LINE: 05-12-2022

UI Engineer

Gurugram, India Part Time 5lpa

[Apply Now](#)

DEAD LINE: 05-12-2022

UI Engineer

Gurugram, India Part Time 5lpa

[Apply Now](#)

DEAD LINE: 05-12-2022

Software Tester

Mumbai, India Full Time 7lpa

[Apply Now](#)

DEAD LINE: 03-01-2023

UI Engineer

Gurugram, India Part Time 5lpa

[Apply Now](#)

DEAD LINE: 05-12-2022

Software Tester

Mumbai, India Full Time 7lpa

[Apply Now](#)

DEAD LINE: 03-01-2023

Software Engineer

Noida, India Full Time 8lpa

[Apply Now](#)

DEAD LINE: 25-11-2022

UI Engineer

[Apply Now](#)



Got any questions?

Our team is always here to help. Send us a message and we'll get back to you shortly.

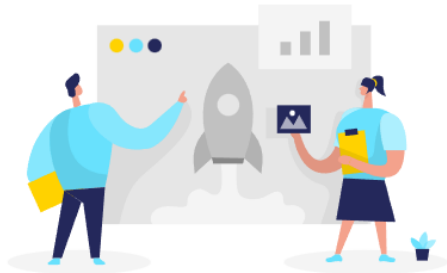
FAQ

Commonly asked questions

Login Page

[Home](#)[Discover](#)[Contacts](#)[Log in](#)[Sign up](#)

Login

[Log in](#)[Forgot your password?](#)

Job Hunter

The most powerful platform
for job seekers as well as
recruiter

Discover

Web design
Development
Hosting

About

Company
Team
Legacy

Careers

Job openings
Employee success
Benefits

Signup Page



[Home](#) [Discover](#) [Contacts](#) [Log in](#)

[Sign up](#)



Sign up

[Create account](#)

Already have an account? [Log in](#) →

Job Hunter

The most powerful platform
for job seekers as well as
recruiter

Discover

Web design
Development
Hosting

About

Company
Team
Legacy



Careers

Job openings
Employee success
Benefits

Jobs Database in IBM cloud

PBP31273.JOBS					Back
					Export to CSV
NAME	LOCATION	TYPE	SALARY	DEADLINE	
Mechanical Engineer	Delhi, India	Full Time	4lpa	16-12-2022	
SOFTWARE ENGINEER	NOIDA, INDIA	FULL TIME	8LPA	25-11-2022	
Software Engineer	Noida, India	Full Time	8lpa	25-11-2022	
Software Engineer	Pune, India	Full Time	13lpa	18-12-2022	
Software Engineer	Noida, India	Full Time	8lpa	25-11-2022	
Software Tester	Mumbai, India	Full Time	7lpa	03-01-2023	
Software Tester	Mumbai, India	Full Time	7lpa	03-01-2023	
Software Tester	Mumbai, India	Full Time	7lpa	03-01-2023	
Software Tester	Mumbai, India	Full Time	7lpa	03-01-2023	
UI Engineer	Gurugram, India	Part Time	5lpa	05-12-2022	
UI Engineer	Gurugram, India	Part Time	5lpa	05-12-2022	
UI Engineer	Gurugram, India	Part Time	5lpa	05-12-2022	

Users database - signup and login details

PBP31273.USERS1				Back
				 Export to CSV 
NAME	PHONE	EMAIL	PASSWORD	
Deepak	9944430260	deepak@gmail.com	12345	
Deepak	994526851	deepak@yahoo.com	12345	
Surya	09566600630	emp1@xyz.com	12345	

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

ADVANTAGE :

- It helps candidates to search the job which perfectly suits them and make them aware of all the job openings.
- It helps recruiters of the company to choose the right candidates for their organizations with appropriate skills.
- Since it is a cloud application, it does not require any installation of software and is portable.

DISADVANTAGE:

- It is costly.
- Uninterrupted internet connection is required for smooth functioning of application

11. CONCLUSION

Recommendation systems have the potential to explore new opportunities for job searchers by enabling them to provide customised recommendations to searchers based on information retrieved from the Internet. They help searchers to instantly find the skills and jobs that closely match with their choices. Moreover, different state-of-the-art algorithms have been developed to recommend products based on users' interactions with their social groups. Therefore, research on embedding social media images within skill and job recommendation systems has gained huge popularity in recent times. This paper presented a review of the skill/job recommendation systems, algorithmic models and filtering techniques based on the academic articles related to this topic.

12. FUTURE SCOPE

The hybrid job recommendation approaches presented combined two or more techniques to overcome the problems that suffer from using each technique separately. For example, while the probability hybrid approaches in paragraph A realised a bidirectional recommendation and tried to cover different selection dimensions, they need to enhance by including more features for individuals and extending by various relational aspects other than trust. Additionally, they only adopted the binary representation with Yes and No when stating user preferences, and it cannot measure the degree of users preferences for each index well, it presented some approaches and systems based on CBF techniques. As mentioned in the CBF, it is limited by the features that are explicitly associated with recommended objects. Therefore, since the applicants' resumes are usually represented by their most important features using some key words, CBF systems cannot distinguish between different keywords meaning. In addition, the problem is usually associated with the pure CBF systems; it cannot recommend jobs that are different from anything the user has seen before. Jobs will be recommended if they are similar to other jobs that the applicant is already interested in. Thus, the applicants have to rate a sufficient number of jobs before a CBF recommender system can really understand the applicant's preferences and present reliable recommendations. For example, the machine learning recommender system builds an automated system to recommend jobs for applicants based on their past job histories. This system uses a classifier that makes a recommendation by training them on content information. It suffered from scalability and data problems.

13. APPENDIX

Adomavicius G, Tuzhilin A (2005). Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. Knowl. Data Eng.* 17(6):734-749. Barbieri N, Costa G, Manco G, Ortale R (2011).

Modeling Item Selection and Relevance for Accurate Recommendations: A Bayesian Approach. In *Proceedings of the fifth ACM conference on Recommender systems (RecSys '11)*, Chicago, Illinois, USA, ACM pp. 21-28. Belkhirat A, Belkhir A, Bouras A (2011).

A New Similarity Measure for the Profiles Management. *UKSim 13th International Conference on Modelling and Simulation*. Cambridge, United Kingdom: IEEE.

Bhargava HK, Sridhar SC (1999). Beyond Spreadsheets: Tools for Building Decision Support Systems. *IEEE Computer* 32(3):31-39. Breaugh JA, Starke M (2000).

Research on Employee Recruitment: So Many Studies, So Many Remaining Questions. *J. Manag.* 26(3):405- 434. Breese JS, Heckerman D, Kadie C (1998).

Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI, ACM, pp. 43-52. Brusilovsky P (2001).

Adaptive hypermedia. *User Model. User Adapt. Interact.* 11(1-2):87-110. Burke R (1999). Integrating Knowledge-Based and Collaborative Filtering Recommender Systems. In *Proceedings of the AAAI Workshop on AI in Electronic Commerce*, Orlando, Florida, USA pp. 69-72. Burke R (2000).

Knowledge-Based Recommender Systems. *Encyclopedia of Library and Information Systems* 69:32. Burke R (2002). Hybrid Recommender Systems: Survey and Experiments. *User Model. User-Adapt. Interact* 12(4):331-370. Burke R (2007).

Hybrid Web Recommender Systems. *The Adaptive Web: Methods and Strategies of Web Personalization* 4321:377-407. Carroll M, Marchington M, Earnshaw J, Taylor S (1999). Recruitment in Small Firms: Processes, Methods and Problems. *Employee Relations* 21(3):236-250. Chen PC (2009).

A Fuzzy Multiple Criteria Decision Making Model in Employee Recruitment. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* 9(7):113-117. Das A, Datar M, Garg A, Rajaram S (2007).

Google News Personalization Scalable Online Collaborative Filtering. In Proceedings of the 16th International Conference on World Wide Web, WWW '07, Banff, Alberta, CANADA, ACM, pp. 271-280. Färber F, Weitzel T, Keim T (2003).

An Automated Recommendation Approach to Selection in Personnel Recruitment. In Proceedings of AMCIS. Tampa, FL, USA, AISel. Fazel-Zarandi M, Fox MS (2010).

Semantic Matchmaking for Job Recruitment: An Ontology Based Hybrid Approach. In Proceedings of the 3rd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web at the 8th International Semantic Web Conference. Washington D.C., USA, CEUR-WS.org. Felfernig A (2005).

Koba4MS: selling complex products and services using knowledge-based recommender technologies. In Proceedings of 7th IEEE International Conference on E-Commerce Technol. Munich, Germany, IEEE pp. 92-100. Felfernig A, Schubert M, Mandl M (2010).

Recommendation and Decision Technologies For Requirements Engineering. RSSE '10. Cape Town, South Africa, ACM pp. 11-15. Ghazanfar MA, Prügell-Bennett A (2010).

Building Switching Hybrid Recommender System Using Machine Learning Classifiers and Collaborative Filtering. IAENG Int. J. Comput. Sci. 37:3. Gunawardana A, Meek C (2009).

A Unified Approach to Building Hybrid Recommender Systems. In Proceedings of the third ACM conference on Recommender systems, New York, NY, USA, ACM pp. 117-124. Hanani U, Shapira B, Shoval P (2001).

Information Filtering: Overview of Issues and Systems. User Modeling and User-Adapted Interaction

Source Code

GitHub & Project Demo Link