1. INTRODUCTION

1.1 Project Overview

Nowadays, job search is a task commonly done on the Internet using job search engine sites like LinkedIn,Indeed, and others. Commonly, a job seeker has two ways to search a job using these sites: 1) doing a querybased on keywords related to the job vacancy that he/she is looking for, or 2) creating and/or updating a professional profile containing data related to his/her education, professional experience, professional skills andother, and receive personalized job recommendations based on this data. Sites providing support to the former case are more popular and have a simpler structure; however, their recommendations are less accurate than those of the sites using profile data.Personalized job recommendation sites implemented a variety of types of recommender systems, such as content-based filtering, collaborative filtering, knowledge-based and hybrid approaches.

1.2 Purpose

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.

2. LITERATURE SURVEY

2.1 Existing Problem

1. Students/Job seekers find their desired job based on their skillset.
2. Integrating Intelligent CHATBOT for job recommendation application.
3. A study of LinkedIn as an Employment Tool for Job seeker & Recruiter.
4. Cloud storage and sharing services.

2.2 References

1. A combined representation learning approach for better job and skill recommendation- Vachik S.Dave Baichuan Zhang

2. A Recommender System for Job Seeking and Recruiting - Yao Lu, Sandy El Helou, Denis Gillet.

3. Job Recommendation based on Job Seeker Skills an Empirical Study - Jorge ValverdeRebaza, Ricardo Puma, Paul

2.3 Problem Statement Definition

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.

3.IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas:

3.2 Ideation & Brainstorming:

3.3 Porposed solution:

Problem Statement (Problem to be solved):

To enable job seekers to find jobs relevant to their skill set.

3.4 Problem Solution fit:

4.REQUIREMENT ANALYSIS

4.1FunctionalRequirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registrationthrough Gmail
FR-2	User Confirmation	Confirmation via Email
FR-3	Chat Bot	A Chat Bot will be there in website to solve userqueries and problems related to applying a job, searchfor a joband muchmore.
FR-4	User Login	Login through Gmail
FR-5	User Search	Exploration of Jobsbased on job filters and skillrecommendations.
FR-6	User Profile	Updation of theuser profile through the logincredentials
FR-7	User Acceptance	Confirmation of the Job.

4.2Non-functional Requirements:

NFR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This application can be used by the job seekers to loginand search for the job based on her Skillsset.
NFR-2	**Security**	This application is secure with separate loginfor JobSeekers as well as Job Recruiters.
NFR-3	**Reliability**	In This application user can able to n number of jobs for free without any limitations and also the enormous job openings will be provided to all thejob seekers without any limitation.
NFR-4	**Performance**	The performance of this application is quicker response and takes lessertime 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 timeof the application is quite fastercompared to anyother application.

5.PROJECT DESIGN

5.1 Data flow diagram:

A Data Flow Diagram (DFD) is a traditional visual representation of theinformation 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 leavesthe system, what changesthe information, and where data is stored.

5.2 Solution and technical architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1& table2

5.3 User Stories:

Use the below templateto list all the user stories for the product.

User type	Functionalrequirements	User storynumber	Task	AcceptanceCriteria	Priority	release
Customer	Registration	USN-1	As a user, I canregister for the application by entering my email, password, and confirming my password.	I can access myaccount / dashboard	High	Sprint-1
		USN-2		I can receive confirmation email & click confirm	High	Sprint-1

		USN-3	As a user, I canregister for the application through online websites	I can register & access the dashboard with onlinewebsite Login	Low	Sprint-2
		USN-4	As a user, I canregister for the application through Gmail	confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	confirm	High	Sprint-1
Customer		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 canable to get personalized job recommendations.	I can receive jobrecommendations	Medium	Sprint-1
Customercareexecutive		USN-7	24/7 chatbotsupport.	24/7 chatbotsupport	High	Sprint-1
admin		USN-8	As an administrator, I can able to viewthe progress andmake required changes in the project	Deploy user specific and personalized jobrecommendations	High	Sprint-1

6 . PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation:

Sprint	FunctionalRequirement	User Story Number	User Story / Task	Priority	Acceptancecriteria	Team Members
Sprint-1	UIDesign	USN-1	As a user, I can see and experience an awesome user interface in the website	Medium	Better Impression about a website	Gokuldass A, Rithisha R
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email,password, and confirming my password.	High	Ican access my account I dashboard	Tariq Jamal A, Mohammed Arshad Z
Sprint-1		USN-3	As a user, I will receive confirmationemail once I have registered for theapplication	High	I can receive confirmation email & click confirm	Gokuldass A, Mohammed Arshad Z
Sprint-1		USN-4	As a user,I can register for theapplication throughFacebook	Low	I can register & access the dashboard with Facebook Login	Mohammed Arshad Z, Rithisha R
Sprint-1		USN-5	As a user,I can register for theapplication through Gmail	Medium	I can receive confirmation email & click confirm	Tariq Jamal A, Rithisha R

Sprint-1	Login	USN-6	As a user,I can log into the applicationby entering email & password	High	Ican access my account I dashboard	Tariq Jamal A, Mohammed Arshad Z
Sprint-1	Flask	USN-7	As a user, Ican access the website in asecond	High	Ican access my account I dashboard	Mohammed Arshad Z, Gokuldass A

Sprint	FunctionalRequirement (Epic)	User Story Number	User Story / Task	Priority	Acceptancecriteria	Team Members
Sprint-1	Dashboard	USN-8	As a user,If I Logged in correctly, I canview my dashboard and I can navigateto any pages which are already listed there.	High	I can accessall the pages/dashboard	Tariq Jamal A
			SubmissionOf Sprint-1			
Sprint-2	User Profile	USN-9	As a user, I can view and update mydetails	Medium	I can modify my details/data	Tariq Jamal A, Gokuldass A, Mohammed Arshad Z
Sprint-2	Database	USN-10	As a user, I can store my details anddata in the website	Medium	I can store my data	Rithisha R, Gokuldass A

Sprint-2	Cloud Storage	USN-11	As a user, I can upload my photo,resumeand much more in the website.	Medium	I can Uploadmy documents anddetails	Tariq Jamal A, Mohammed Arshad Z
Sprint-2	Chatbot	USN-12	As a user, I can ask the Chatbot aboutlatest job openings, which willhelpme and show the recentjob openings based on my profile.	High	I can knowthe recent jobopenings	Tariq Jamal A, Gokuldass A
Sprint-2	Identity-Aware	USN-13	As a User, I can access my account by entering by correct logincredentials. My user credentials is only displayed to me.	High	I can have my account safely	Rithisha R, Mohammed Arshad Z
			Submissionof Sprint-2			

Sprint	FunctionalRequirement (Epic)	User Story Number	User Story / Task	Priority	Acceptancecriteria	Team Members
Sprint-3	Sendgrid service	USN-14	As a user, I can get a notification or mail abouta job opening with the helpof sendgrid service.	Medium	I can get a notification in asecond.	Mohammed Arshad Z, Gokuldass A

Sprint-3	Learning Resource	USN-15	As a user, I can learn the course andI will attain the skills which will be useful for developing my technical skills.	High	I can gainthe knowledge and skills	Tariq Jamal A, Rithisha R,
Sprint-3	Docker	USN-16	As a user, I can access the website in anydevice	High	I can accessmy account in anydevice	Tariq Jamal A, Gokuldass A
Sprint-3	Kubernates	USN-17	As a user, I can access thewebsite in any device	High	I can accessmy account in anydevice	Tariq Jamal A, Gokuldass A
Sprint-3	Deployment in cloud	USN-18	As a user, I can access thewebsite in any device	High	I can accessmy account in anydevice	Rithisha R, Mohammed Arshad Z, Gokuldass A
Sprint-3	Technical support	USN-19	As a user, I can get a customercaresupport from the website which will solve my queries.	Medium	Ican tackle my problem&queries.	Tariq Jamal A, Rithisha R
			Submission of Sprint-3			

Sprint-4	Unit Testing	USN-15	As a user, I can access the web site withoutany interruption	High	I can accessthe website withoutany interruption	TariqJamalA, MohammedArshadZ, GokuldassA, RithishaR
Sprint-4	Integrationtesting	USN-16	As a user, I can access the web site withoutany interruption	High	I can accessthe websitewithout any interruption	TariqJamalA, Moha

						mmedArshadZ, GokuldassA, RithishaR

Sprint	FunctionalRequirement (Epic)	User Story Number	User Story / Task	Priority	Acceptancecriteria	Team Members
Sprint-4	Systemtesting	USN-17	As a user, I can access the website withoutany interruption	High	I can accessthe websitewithout any interruption	TariqJamalA, MohammedArshad

						Z, Gokuldass A, Rithisha R
Sprint-4	Correction	USN-18	As a user, I can access the website withoutany interruption	High	I can accessthe websitewithout any interruption	Tariq Jamal A, Mohammed Arshad Z, Gokuldass A, Rithish

						aR
Sprint-4	Acceptancetesting	USN-19	As a user, I can access the website withoutany interruption	High	I can accessthe websitewithout any interruption	TariqJamalA, MohammedArshadZ, GokuldassA, RithishaR
			Submission of Sprint-4			

6.2 Sprint Delivery Schedule:

Project Tracker, Velocity & Burndown Chart

Sprint	Total StoryPoints	Duration	Sprint Start Date	Sprint End Date(Planned)	Story PointsCompleted (as on Planned End Date)	Sprint ReleaseDate (Actual)
Sprint-1	20	6 Days	2022-10-30	2022-11-04	20	2022-11-04
Sprint-2	20	5 Days	2022-11-05	2022-11-09	20	2022-11-09
Sprint-3	20	6 Days	2022-11-10	2022-11-15	20	2022-11-15
Sprint-4	20	4 Days	2022-11-16	2022-11-19	20	2022-11-19

6.3 Reports from JIRA:

Average velocity = sprint duration / velocity
= 20 / 10 = 2

7. CODING & SOLUTIONING

7.1 Feature 1:
Create Account:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Register {% endblock title %}

{% block heading %}
   Create an account with your credentials
{% endblock heading %}

{% block body %}
   <form method="post">
      <div class="mb-3">
      <label for="username" class="form-label">Username</label>
      <input type="text" name="username" class="form-control"
id="username" placeholder="Enter username" minlength="3"
maxlength="15" required>
    </div>
    <div class="row">
      <div class="col-md-6">
        <div class="mb-3">
          <label for="first_name" class="form-label">First name</label>
          <input type="text" name="first_name" class="form-control"
id="first_name" placeholder="Enter first name" minlength="3"
maxlength="15" required>
        </div>
      </div>
      <div class="col-md-6">
        <div class="mb-3">
          <label for="last_name" class="form-label">Last name</label>
          <input type="text" name="last_name" class="form-control"
id="last_name" placeholder="Enter last name" minlength="1"
maxlength="15" required>
        </div>
      </div>
    </div>
    <div class="mb-3">
      <label for="email" class="form-label">Email</label>
      <input type="email" name="email" class="form-control" id="email"
placeholder="Enter email" required>
    </div>
    <div class="mb-3">
      <label for="phone_number" class="form-label">Phone
number</label>
      <div class="row">
        <div class="col-md-2 col-sm-3 col-4 phno_code">
          <input type="number" value="+91 " readonly placeholder="+91"
class="form-control">
        </div>
        <div class="col-md-10 col-sm-9 col-8 phno">
          <input type="number" name="phone_number" class="form-
control" id="phone_number" placeholder="Enter phone number"
required>
        </div>
      </div>
    </div>
    <div class="mb-3">
      <label for="interest" class="form-label">Interest</label>
      <input type="text" name="interest" class="form-control"
id="interest" placeholder="Enter interest" required>
   </div>
    <div class="row">
      <div class="col-md-6">
        <div class="mb-3">
          <label for="state" class="form-label">State</label>
          <input type="text" name="state" class="form-control" id="state"
placeholder="Enter state" minlength="3" maxlength="15" required>
        </div>
      </div>
      <div class="col-md-6">
        <div class="mb-3">
          <label for="city" class="form-label">City</label>
          <input type="text" name="city" class="form-control" id="city"
placeholder="Enter city" minlength="3" maxlength="15" required>
        </div>
      </div>
    </div>
    <div class="mb-3">
      <label for="password" class="form-label">Password</label>
      <input type="password" name="password" class="form-control"
id="password" placeholder="Enter password" minlength="3"
maxlength="15" required>
    </div>
    <p class="mt-3">Already have an account <a href="{{ url_for('login')
}}">login</a></p>
    <button type="submit" class="btn btn-primary">Register</button>
  </form>
{% endblock body %}
```

User Data:

```
{% extends 'layout/app.html' %}

{% block title %} Profile Page {% endblock title %}
<!-- Page content  -->
{% block body %}
```

```
<div class="container my-3 p-5 shadow-sm">
  <div class="d-flex justify-content-between">
    {% if error %}
      <div class="alert alert-primary" role="alert">
        {{ error }}
      </div>
    {% endif %}
    <h2>User Data</h2>
    <div>
      <button type="button" class="btn btn-primary btn-sm me-2" data-bs-toggle="modal" data-bs-target="#resetpassword">Reset
        Password</button>
      <!-- Modal -->
      <div class="modal fade" id="resetpassword" tabindex="-1" aria-labelledby="resetpasswordLabel"
        aria-hidden="true">
        <div class="modal-dialog modal-sm">
          <div class="modal-content">
            <div class="modal-header">
              <h5 class="modal-title" id="resetpasswordLabel">Reset Password</h5>
              <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <form action="/profile" method="post">
              <div class="modal-body">
                <label for="password" class="form-label">Password</label>
                <input type="password" class="form-control" name="password" \>
                <label for="confirm_password" class="form-label">Confirm Password</label>
                <input type="password" class="form-control" name="confirm_password" \>
              </div>
              <div class="modal-footer">
                <button type="button" class="btn btn-sm btn-secondary" data-bs-dismiss="modal">Close</button>
                <input type="submit" class="btn btn-sm btn-primary" name="forgot_password" value="Save changes">
              </div>
            </form>
          </div>
        </div>
      </div>
      <a class="btn btn-outline-primary btn-sm" href="{{
url_for('update_profile') }}">Update Profile</a>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div>
        <label for="name" class="form-label">Name</label>
        <input type="text" class="form-control" name="name" value="{{
user['user']['NAME'] }}" readonly>
      </div>
   </div>
    <div class="col-md-6">
      <div>
        <label for="username" class="form-label">Username</label>
        <input type="text" class="form-control" name="username"
value="{{ user['user']['USERNAME'] }}"
          readonly>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div>
        <label for="email" class="form-label">email</label>
        <input type="text" class="form-control" name="email" value="{{
user['user']['EMAIL'] }}" readonly>
      </div>
    </div>
    <div class="col-md-6">
      <div>
        <label for="phone_number" class="form-label">Phone
number</label>
        <input type="text" class="form-control" name="phone_number"
value="{{ user['user']['PHNO'] }}"
          readonly>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div>
        <label for="interest" class="form-label">Interest</label>
        <input type="text" class="form-control" name="interest"
value="{{user['user']['INTEREST']}}" readonly>
      </div>
    </div>
    <div class="col-md-3">
      <div>
        <label for="state" class="form-label">State</label>
        <input type="text" class="form-control" name="state"
value="{{user['user']['STATE']}}" readonly>
     </div>
    </div>
    <div class="col-md-3">
      <div>
        <label for="city" class="form-label">City</label>
        <input type="text" class="form-control" name="city"
value="{{user['user']['CITY']}}" readonly>
      </div>
    </div>
  </div>
  <h2 class="text-primary my-2">Education Details</h2>
  {% for education in user['educations'] %}
    <h4>{{ education['COLLEGE_NAME'] }}</h4>
    <p class="text-secondary m-0">{{ education['DEGREE'] }}</p>
   <p class="text-secondary">Grade: {{ education['GRADE'] }}</p>
  {% endfor %}
  <a class="btn btn-primary mt-4" href="{{ url_for('add_education')
}}">Add Education</a>
</div>
{% endblock body %}
```

Reset Password:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Reset Password {% endblock title %}
{% block heading %}
  Reset Password
{% endblock heading %}

{% block body %}
  <p>Forgotten your password? Enter your email address below, and we'll instruction for setting a new password</p>

  <form method="post">
    <div class="mb-3">
      <label for="email" class="form-label">Email</label>
      <input type="email" class="form-control" name="email" id="email">
    </div>
    <input type="submit" class="btn btn-primary" value="Send email">
  </form>
{% endblock body %}
```

Reset Password Completed:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Password Resetted {% endblock title %}


{% block body %}
  <h3 class="text-primary">Password reset complete</h3>
  <p>Your password has been set. You may go ahead and log in now.</p>

  <a href="{% url 'login' %}" class="btn btn-primary">Login</a>
{% endblock body %}
```

7.2 Feature 2:

Post A Job:

```
{% extends 'layout/app.html' %}

{% block title %} Jobs {% endblock title %}
```

```
<!-- Page content  -->
{% block body %}

    <div class="container py-3">
        <div class="row">
            <div class="col-md-6">
                <h3 class="text-primary">Jobs</h3>
            </div>
            <div class="col-md-6">
                <form action="{{url_for('jobs')}}" method="post">
                    <div class="d-flex justify-content-center align-items-center">
                        <input type="text" class="form-control" name="search_query"
placeholder="Enter...">
                        <button class="btn btn-primary"
type="submit">Search</button>
                    </div>
                </form>
            </div>
        </div>
        <div class="row my-4 py-3">
            {% for job in jobs %}
                <div class="col-md-5 shadow rounded p-3 my-3 mx-4">
                    <div class="d-flex justify-content-between align-items-center">
                        <h2 class="text-dark fw-bold">{{ job['title'] }}</h2>
                    </div>
                    <p class="text-secondary">Company: {{
job['company']['display_name'] }}</p>
                    <span class="text-secondary">Locations:
                        {% for location in job['location']['area'] %}
                            {{location}},
                        {% endfor %}
                    </span>
                    <p class="text-secondary">Salary: {{ job['salary_max'] }}</p>
                    <hr>
                    <p class="text-secondary">Job type: {{ job['contract_type']
}}</p>
                    <h4>Description</h4>
                    <p class="text-secondary">{{ job['description'] }}</p>
                    <a href="{{ job['redirect_url'] }}" class="btn btn-primary"
target="_blank">Apply</a>
                </div>
            {% endfor %}
```

```
      </div>
   </div>
{% endblock body %}
```

Jobs:

```
{% extends 'layout/app.html' %}

<!-- Page content  -->
{% block body %}

   <div class="container py-3">
      <div class="row">
         <div class="col">
            <a class="btn btn-primary" href="{{ url_for('jobs') }}">Back</a>
         </div>
      </div>
      <div class="row my-3 p-3 shadow-sm rounded">
         <div class="col-md-12">
            <h2 class="fw-bold">{{ job["TITLE"] }}</h2>
            <h5 class="text-secondary">{{ job["COMPANY_NAME"] }}</h5>
            <p class="text-secondary">{{ job["LOCATION"] }}</p>
            <div class="d-flex" >
               <a class="btn btn-primary me-2" href="/apply_job/{{ job['ID']
}}">Apply job</a>
               <a class="btn btn-outline-primary">Save job</a>
            </div>
            <hr>
            <h4 class="fw-bold">Job details</h4>
            <h5>Job type</h5>
            <P class="text-secondary">{{ job["JOB_TYPE"] }}</P>
            <h5>Experience</h5>
            <P class="text-secondary">{{ job["EXPERIENCE"] }}</P>
            <h5>Salary</h5>
            <P class="text-secondary">{{ job["SALARY"] }}</P>
            <h5>Vacancy</h5>
            <P class="text-secondary">{{ job["VACANCY"] }}</P>
            <h5>Description</h5>
            <p class="text-secondary">{{ job["DESCRIPTION"] }}</p>
            <h5>Education</h5>
            <p class="text-secondary">{{ job["EDUCATION"] }}</p>
            <h5>Responsibility</h5>
            <p class="text-secondary">{{ job["RESPONSIBILITY"] }}</p>
```

```
                <h5>Application Deadline {{ job["DEADLINE"] }}</h5>
                <p class="text-secondary">Posted on {{ job["POSTED_ON"] }}</p>
            </div>

        </div>
    </div>
{% endblock body %}
```

Added Education:

```
{% extends 'layout/app.html' %}

{% block title %} Add Education {% endblock title %}

<!-- Page content  -->
{% block body %}
<div class="container my-4 shadow rounded-3 px-5 py-3 border border-1
border-light">
    <h2 class="text-white text-center bg-primary py-2 rounded">Add
Education</h2>
    <form action="/add_education" method="post">
        <div class="row">
            <div class="col-md-12">
                <div class="mb-3">
                    <label for="college_name" class="form-label">College
name</label>
                    <input type="text" name="college_name" class="form-control"
id="college_name" placeholder="Enter college name">
                </div>
            </div>
        </div>
        <div class="row">
            <div class="col-md-6">
                <div class="mb-3">
                    <label for="degree" class="form-label">Degree</label>
                    <input type="text" name="degree" class="form-control"
id="degree" placeholder="Enter degree">
                </div>
            </div>
            <div class="col-md-6">
                <div class="mb-3">
```

```
          <label for="grade" class="form-label">Grade</label>
          <input type="text" name="grade" class="form-control"
id="grade" placeholder="Enter grade">
        </div>
      </div>
    </div>
    <button type="submit" class="btn btn-primary">Add
Education</button>
  </form>
</div>
{% endblock body %}
```

8. TESTING

8.1 Test Cases:

8.2 User Acceptance Testing:

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Skills and Job Recommendation project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	8	3	3	3	17
Duplicate	1	1	3	1	6
External	2	3	0	1	6
Fixed	12	2	4	19	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	23	14	11	25	75

3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	6	0	1	6
ClientApplication	49	0	1	49
Security	2	0	2	2
OutsourceShipping	3	0	1	3
Exception Reporting	7	0	1	7
Final ReportOutput	4	0	1	4
VersionControl	3	0	0	3

9. RESULTS:

9.1 PERFORMANCE METRICS:

The project has been completed as we expected. We ensured that Database was designed and well connected to our project. The Expected results were gotten.

10. ADVANTAGES & DISADVANTAGES:

ADVANTAGES:

- Person who looks for a job can easily find a suitable job based on their skill set.
- Person can check their eligibility by attending eligibility test.
- Most of the Recruiters find the suitable person based on the scores they have gotten in the eligibility.

DISADVANTAGES :

- Person Job May get technical difficulty while taking the eligibility
- Job seeker may have trouble to contact recruiters directly.

11. CONCLUSION

- The application has been developed to make job search easier .
- The application that we have developed is user friendly .
- User can find a job based on their skillset in the short period of time.

- The jobseeker certainly get benefit by using this application. In the addition,Chatbot Has been implemented with the help of IBM whatson .
- The chatbot helps jobseeker and organization when they experience the difficulties.

12. FUTURE SCOPE

The linked in the wellknown application to find a job and stay connected with professional and organization. The job seekers and organization use linked in to find a job. In the future , There are lots of possibilities to enhance our project similar to linked in.

13. APPENDIX

Source Code:

FRONT END:

Authentication:

login.html:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Login {% endblock title %}

{% block heading %}
  Login with your credentials
{% endblock heading %}

{% block body %}
  <form method="post">
    {% with messages = get_flashed_messages() %}
      {% if messages %}
        {% for message in messages %}
          <h4 class="text-danger">{{ message }}</h4>
        {% endfor %}
      {% endif %}
    {% endwith %}
    <div class="mb-3">
      <label for="username" class="form-label">Username</label>
      <input type="text" name="username" class="form-control"
id="username" placeholder="Enter username">
    </div>
    <div class="mb-3">
      <label for="password" class="form-label">Password</label>
      <input type="password" name="password" class="form-control"
id="password" placeholder="Enter password">
    </div>
    <p class="mt-3">Don't have an account <a href="{{ url_for('register')
}}">register</a></p>
    <button type="submit" class="btn btn-primary">Login</button>
  </form>
{% endblock body %}
```

register.html:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Register {% endblock title %}

{% block heading %}
  Create an account with your credentials
{% endblock heading %}

{% block body %}
  <form method="post">
    <div class="mb-3">
      <label for="username" class="form-label">Username</label>
      <input type="text" name="username" class="form-control"
id="username" placeholder="Enter username" minlength="3"
maxlength="15" required>
    </div>
    <div class="row">
      <div class="col-md-6">
        <div class="mb-3">
          <label for="first_name" class="form-label">First name</label>
```

```
                <input type="text" name="first_name" class="form-control"
id="first_name" placeholder="Enter first name" minlength="3"
maxlength="15" required>
            </div>
          </div>
          <div class="col-md-6">
            <div class="mb-3">
                <label for="last_name" class="form-label">Last name</label>
                <input type="text" name="last_name" class="form-control"
id="last_name" placeholder="Enter last name" minlength="1"
maxlength="15" required>
            </div>
          </div>
        </div>
        <div class="mb-3">
          <label for="email" class="form-label">Email</label>
          <input type="email" name="email" class="form-control" id="email"
placeholder="Enter email" required>
        </div>
        <div class="mb-3">
          <label for="phone_number" class="form-label">Phone
number</label>
          <div class="row">
            <div class="col-md-2 col-sm-3 col-4 phno_code">
                <input type="number" value="+91 " readonly placeholder="+91"
class="form-control">
            </div>
            <div class="col-md-10 col-sm-9 col-8 phno">
                <input type="number" name="phone_number" class="form-
control" id="phone_number" placeholder="Enter phone number"
required>
            </div>
          </div>
        </div>
      <div class="mb-3">
          <label for="interest" class="form-label">Interest</label>
          <input type="text" name="interest" class="form-control"
id="interest" placeholder="Enter interest" required>
        </div>
        <div class="row">
          <div class="col-md-6">
            <div class="mb-3">
          <label for="state" class="form-label">State</label>
          <input type="text" name="state" class="form-control" id="state"
placeholder="Enter state" minlength="3" maxlength="15" required>
        </div>
      </div>
      <div class="col-md-6">
        <div class="mb-3">
          <label for="city" class="form-label">City</label>
          <input type="text" name="city" class="form-control" id="city"
placeholder="Enter city" minlength="3" maxlength="15" required>
        </div>
      </div>
    </div>
    <div class="mb-3">
      <label for="password" class="form-label">Password</label>
      <input type="password" name="password" class="form-control"
id="password" placeholder="Enter password" minlength="3"
maxlength="15" required>
    </div>
    <p class="mt-3">Already have an account <a href="{{ url_for('login')
}}">login</a></p>
    <button type="submit" class="btn btn-primary">Register</button>
  </form>
{% endblock body %}
```

verification_email_send.html:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Account verification {% endblock title %}


{% block body %}
  <h3 class="text-primary">Verification email send</h3>
  <p>Please confirm your email address to complete the registration</p>
{% endblock body %}
```

account_activation_success.html:

```
{% extends 'auth_layout/app.html' %}

{% block title %} Account verification {% endblock title %}
{% block body %}
    {% if is_success %}
        <p><b>{{ username }}</b> your account has been activated
successfully.</p>
        <p>You can login now</p>
        <a class="btn btn-primary" href="{{ url_for('login') }}">Login</a>
    {% else %}
        <h2 class="text-danger">Something went wrong token is not
validated</h2>
    {% endif %}
{% endblock body %}
```

Layout:

app.html:

```
<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <!-- Bootstrap CSS -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css" rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuC
OmLASjC" crossorigin="anonymous">

    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/6.2.0/css/all.min.css">

    <title>{% block title %} {% endblock title %}</title>

    <style>
        .main {
            background-image: linear-gradient(rgba(0, 0, 0, 0.6), rgba(83, 83, 83,
0.3)), url('https://images.pexels.com/photos/838413/pexels-photo-
838413.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1');
```

```
      background-size: cover;
      background-position: center;
      min-height: 90vh;
      width: 100%;
    }

    .apply {
      background: url('static/images/background.png')no-repeat;
      background-size: cover;
      height: 110vh;
      width: 100%;
    }

    .apply .card {
     background-color: rgba(4, 26, 100, 0.801);
      color: white;
    }

    .cir1 img {
      border-radius: 50%;
    }
  </style>

</head>

<body onload="myFunction()">
  {% include 'layout/header.html' %}

  <!-- Page content  -->
  {% block body %}

  {% endblock body %}

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bun
dle.min.js"
    integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtI
axVXM"
    crossorigin="anonymous"></script>
    <script>
      window.watsonAssistantChatOptions = {
       integrationID: "14c61a15-4802-4e67-b6b6-beeff792ff88", // The ID of this integration.
       region: "jp-tok", // The region your integration is hosted in.
       serviceInstanceID: "a9372ca4-6fba-4a6b-b71c-8a9db5d9f79c", // 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>
</body>

</html>
```

header.html:

```
<nav class="navbar navbar-expand-lg navbar-light bg-primary shadow-sm">
  <div class="container-fluid">
    <a class="navbar-brand text-white fw-bold fs-3" href="{{ url_for('home') }}">Job Portal</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav me-auto mb-2 mb-lg-0">
        <li class="nav-item">
          <a class="nav-link active text-white fs-5" aria-current="page" href="{{ url_for('home') }}">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link active text-white fs-5" aria-current="page" href="{{ url_for('jobs') }}">Find jobs</a>
        </li>
      </ul>
      <div>
        {% if session['is_loggedin'] %}
          <div class="dropdown">
            <button class="btn btn-light dropdown-toggle" type="button"
id="dropdownMenuButton1" data-bs-toggle="dropdown" aria-
expanded="false">
              <span class="text-primary">{{ session['email'] }}</sp>
            </button>
            <ul class="dropdown-menu" aria-
labelledby="dropdownMenuButton1">
              <li><a class="dropdown-item" href="{{ url_for('profile')
}}">Profile</a></li>
              <li><a class="dropdown-item" href="{{ url_for('logout')
}}">Logout</a></li>
            </ul>
          </div>
        {% else %}
          <a href="" class="btn btn-light me-2">Register</a>
          <a href="" class="btn btn-outline-light">Login</a>
        {% endif %}
      </div>
    </div>
  </div>
</nav>
```

index.html:

```
{% extends 'layout/app.html' %}

{% block title %} Home Page {% endblock title %}
<!-- Page content  -->
{% block body %}
<section class="main py-5 ">
  <div class="container py-5">
    <div class="row">
      <div class="col-lg-7 mb-5">
        <h1 class="text-primary fs-1 fs-bolder">Find The Most Exciting
Startup Jobs</h1>
      </div>
      <div class="col-lg-7 mt-3">
```

```
        <form action="{{url_for('jobs')}}" method="post">
          <div class="row">
              <div class="col-md-8">
                <input type="text" class="form-control py-2"
name="search_query" id="" placeholder="find jobs...">
              </div>
              <div class="col-md-4">
                <button class="btn btn-primary py-2" type="submit">Find
Job</button>
              </div>
          </div>
        </form>
      </div>

   </div>
</section>
<section class="job py-5">
  <div class="container py-5">
    <div class="row py-5">
      <div class="col-lg-7">
        <h4 class=" pt-5">What We Are Doing</h4>
        <h1 class="text-primary">24K Talented Peoples Are Getting
Jobs</h1>
        <p class="my-3 text-secondary">Lorem ipsum dolor sit amet
consectetur adipisicing elit. Doloremque
          totam, accusamus laboriosam quidem quo nulla optio illum
magni deleniti facilis iste! Cupiditate
          voluptatibus impedit doloribus sunt consequuntur. Nam error
dolorum possimus veritatis debitis velit
          voluptas commodi quidem, quisquam rem enim inventore sit
vero nobis facilis nihil aut eius
          consectetur illo.</p>
        <button class="btn btn-primary me-5">Post A Job</button>
      </div>
      <div class="col lg-5">
        <img src="static/images/mens.jpg" class="img-fluid" alt="">
      </div>
    </div>
  </div>
</section>

<section class="apply py-5">
```

```
  <div class="container text-white text-center py-5">
    <p class="red">Apply Process</p>
    <h1>How It Works</h1>
    <div class="row py-5">
      <div class="col-lg-4">
        <div class="card p-3">
          <div class="card-body text-center">
            <div class="cir1 pt-3 mb-3">
              <img src="static/images/cate4.jpg" class="img-fluid" alt="">
            </div>
            <h2>1.Search a Job</h2>
            <p class="py-3">Sorem spsum dolor sit amsectetur adipisclit,
seddo eiusmod tempor
              incididuntutlaborea.
            </p>
          </div>
        </div>
      </div>
      <div class="col-lg-4">
        <div class="card p-3">
          <div class="card-body text-center">
            <div class="cir1 pt-3 mb-3">
              <img src="static/images/cate3.JPG" class="img-fluid" alt="">
            </div>
            <h2>1.Apply for a Job</h2>
            <p class="py-3">Sorem spsum dolor sit amsectetur adipisclit,
seddo eiusmod tempor
              incididuntutlaborea.
            </p>
          </div>
        </div>
      </div>
      <div class="col-lg-4">
        <div class="card p-3">
          <div class="card-body text-center">
            <div class="cir1 pt-3 mb-3">
              <img src="static/images/cate5.JPG" class="img-fluid" alt="">
            </div>
            <h2>1.Get hired by MNC's</h2>
            <p class="py-3">Sorem spsum dolor sit amsectetur adipisclit,
seddo eiusmod tempor
              incididunutlaborea.
```

```
                </p>
              </div>
            </div>
          </div>
        </div>
    </div>
</section>
<footer class="w-100 p-3 bg-dark">
    <p class="text-center m-0 text-white">@2022 Job Portal All rights
reserved.</p>
</footer>

{% endblock body %}
```

jobs.html:

```
{% extends 'layout/app.html' %}

{% block title %} Jobs {% endblock title %}
<!-- Page content  -->
{% block body %}

    <div class="container py-3">
        <div class="row">
          <div class="col-md-6">
            <h3 class="text-primary">Jobs</h3>
          </div>
          <div class="col-md-6">
            <form action="{{url_for('jobs')}}" method="post">
              <div class="d-flex justify-content-center align-items-center">
                <input type="text" class="form-control" name="search_query"
placeholder="Enter...">
                <button class="btn btn-primary"
type="submit">Search</button>
              </div>
            </form>
          </div>
        </div>
        <div class="row my-4 py-3">
          {% for job in jobs %}
            <div class="col-md-5 shadow rounded p-3 my-3 mx-4">
              <div class="d-flex justify-content-between align-items-center">
                <h2 class="text-dark fw-bold">{{ job['title'] }}</h2>
              </div>
              <p class="text-secondary">Company: {{
job['company']['display_name'] }}</p>
              <span class="text-secondary">Locations:
                {% for location in job['location']['area'] %}
                  {{location}},
                {% endfor %}
              </span>
              <p class="text-secondary">Salary: {{ job['salary_max'] }}</p>
              <hr>
              <p class="text-secondary">Job type: {{ job['contract_type']
}}</p>
              <h4>Description</h4>
              <p class="text-secondary">{{ job['description'] }}</p>
              <a href="{{ job['redirect_url'] }}" class="btn btn-primary"
target="_blank">Apply</a>
            </div>
          {% endfor %}
        </div>
    </div>
{% endblock body %}
```

profile.html:

```
{% extends 'layout/app.html' %}

{% block title %} Profile Page {% endblock title %}
<!-- Page content  -->
{% block body %}
<div class="container my-3 p-5 shadow-sm">
  <div class="d-flex justify-content-between">
    {% if error %}
      <div class="alert alert-primary" role="alert">
        {{ error }}
      </div>
    {% endif %}
    <h2>User Data</h2>
    <div>
      <button type="button" class="btn btn-primary btn-sm me-2" data-
bs-toggle="modal" data-bs-target="#resetpassword">Reset
        Password</button>
      <!-- Modal -->
      <div class="modal fade" id="resetpassword" tabindex="-1" aria-labelledby="resetpasswordLabel"
        aria-hidden="true">
        <div class="modal-dialog modal-sm">
          <div class="modal-content">
            <div class="modal-header">
              <h5 class="modal-title" id="resetpasswordLabel">Reset Password</h5>
              <button type="button" class="btn-close" data-bs-dismiss="modal" aria-label="Close"></button>
            </div>
            <form action="/profile" method="post">
              <div class="modal-body">
                <label for="password" class="form-label">Password</label>
                <input type="password" class="form-control" name="password" \>
                <label for="confirm_password" class="form-label">Confirm Password</label>
                <input type="password" class="form-control" name="confirm_password" \>
              </div>
              <div class="modal-footer">
                <button type="button" class="btn btn-sm btn-secondary" data-bs-dismiss="modal">Close</button>
                <input type="submit" class="btn btn-sm btn-primary" name="forgot_password" value="Save changes">
              </div>
            </form>
          </div>
        </div>
      </div>
      <a class="btn btn-outline-primary btn-sm" href="{{ url_for('update_profile') }}">Update Profile</a>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div>
        <label for="name" class="form-label">Name</label>
        <input type="text" class="form-control" name="name" value="{{ user['user']['NAME'] }}" readonly>
      </div>
    </div>
    <div class="col-md-6">
      <div>
        <label for="username" class="form-label">Username</label>
        <input type="text" class="form-control" name="username"
value="{{ user['user']['USERNAME'] }}"
          readonly>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div>
        <label for="email" class="form-label">email</label>
        <input type="text" class="form-control" name="email" value="{{
user['user']['EMAIL'] }}" readonly>
      </div>
    </div>
   <div class="col-md-6">
      <div>
        <label for="phone_number" class="form-label">Phone
number</label>
        <input type="text" class="form-control" name="phone_number"
value="{{ user['user']['PHNO'] }}"
          readonly>
      </div>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <div>
        <label for="interest" class="form-label">Interest</label>
        <input type="text" class="form-control" name="interest"
value="{{user['user']['INTEREST']}}" readonly>
      </div>
    </div>
    <div class="col-md-3">
      <div>
        <label for="state" class="form-label">State</label>
        <input type="text" class="form-control" name="state"
value="{{user['user']['STATE']}}" readonly>
```

```
      </div>
    </div>
    <div class="col-md-3">
      <div>
        <label for="city" class="form-label">City</label>
        <input type="text" class="form-control" name="city" value="{{user['user']['CITY']}}" readonly>
      </div>
    </div>
  </div>
  <h2 class="text-primary my-2">Education Details</h2>
  {% for education in user['educations'] %}
   <h4>{{ education['COLLEGE_NAME'] }}</h4>
    <p class="text-secondary m-0">{{ education['DEGREE'] }}</p>
    <p class="text-secondary">Grade: {{ education['GRADE'] }}</p>
  {% endfor %}
  <a class="btn btn-primary mt-4" href="{{ url_for('add_education') }}">Add Education</a>
</div>
{% endblock body %}
```

update_profile.html:

```
{% extends 'layout/app.html' %}

{% block title %} Profile Page {% endblock title %}
<!-- Page content  -->
{% block body %}
<div class="container my-3 p-5 shadow-sm">
  <div class="d-flex justify-content-between">
    {% if error %}
      <div class="alert alert-primary" role="alert">
        {{ error }}
      </div>
    {% endif %}
    <h2 class="w-100 bg-primary text-center py-2 rounded text-white">User Data</h2>
  </div>
  <form action="/update_profile" method="post">
    <div class="row">
      <div class="col-md-6">
        <div>
```

```
          <label for="name" class="form-label">Name</label>
          <input type="text" class="form-control" name="name" value="{{
user['NAME'] }}" readonly>
        </div>
      </div>
      <div class="col-md-6">
        <div>
          <label for="username" class="form-label">Username</label>
          <input type="text" class="form-control" name="username"
value="{{ user['USERNAME'] }}"
            readonly>
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <div>
          <label for="email" class="form-label">email</label>
          <input type="text" class="form-control" name="email" value="{{
user['EMAIL'] }}" readonly>
        </div>
      </div>
      <div class="col-md-6">
        <div>
          <label for="phone_number" class="form-label">Phone
number</label>
          <input type="text" class="form-control" name="phno" value="{{
user['PHNO'] }}">
        </div>
      </div>
    </div>
    <div class="row">
      <div class="col-md-6">
        <div>
          <label for="interest" class="form-label">Interest</label>
          <input type="text" class="form-control" name="interest"
value="{{user['INTEREST']}}" >
        </div>
      </div>
      <div class="col-md-3">
        <div>
          <label for="state" class="form-label">State</label>
          <input type="text" class="form-control" name="state"
value="{{user['STATE']}}">
        </div>
      </div>
      <div class="col-md-3">
        <div>
        <label for="city" class="form-label">City</label>
          <input type="text" class="form-control" name="city"
value="{{user['CITY']}}">
        </div>
      </div>
    </div>
    <button type="submit" class="btn btn-primary my-
3">Update</button>
  </form>
</div>
{% endblock body %}
```

BACKEND :

app.py:

```
from flask import Flask ,render_template, request, redirect, url_for, session,
g, flash
from passlib.hash import sha256_crypt
from dotenv import load_dotenv
from datetime import timedelta
from flask_mail import Mail, Message
from itsdangerous import URLSafeTimedSerializer, SignatureExpired
from datetime import datetime
import ibm_db
import os
import requests
import json


load_dotenv()

app = Flask(__name__)

DATABASE_NAME = os.getenv("DATABASE_NAME")
HOST_NAME = os.getenv("HOST_NAME")
PORT_NUMBER = os.getenv("PORT_NUMBER")
```

```
USER_ID = os.getenv("USER_ID")
PASSWORD = os.getenv("PASSWORD")

# IBM cloud connection string
conn =
ibm_db.connect(f"DATABASE={DATABASE_NAME};HOSTNAME={HOST_NA
ME};PORT={PORT_NUMBER};SECURITY=SSL;SSLServerCertificate=DigiCer
tGlobalRootCA.crt;UID={USER_ID};PWD={PASSWORD}",'','')

# secret key
app.secret_key = os.getenv('SECRET_KEY')

# configuration of mail
app.config['MAIL_SERVER']='smtp.gmail.com'
app.config['MAIL_PORT'] = 465
app.config['MAIL_USERNAME'] = os.getenv('MAIL_USERNAME')
app.config['MAIL_PASSWORD'] = os.getenv('MAIL_PASSWORD')
app.config['MAIL_USE_TLS'] = False
app.config['MAIL_USE_SSL'] = True

mail = Mail(app)
s = URLSafeTimedSerializer('Thisisasecret!')

app.permanent_session_lifetime = timedelta(days=30)

@app.route("/")
def home():
    if g.email:
        return render_template('index.html')
    else:
        return redirect(url_for('login'))

@app.before_request
def before_request():
    g.email = None

    if 'email' in session:
        g.email = session['email']


# auth routes

@app.route('/login', methods=['POST', "GET"])
def login():
    if request.method == 'POST':
        username = request.form.get('username')
        password = request.form.get('password')

        sql = "select * from FVS49663.USER where username = ?"
        stmt = ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt, 1, username)
        ibm_db.execute(stmt)
        user = ibm_db.fetch_assoc(stmt)
        if user['IS_VERIFIED']:
            if user:
                print(user)
                print(user['ID'])
                user_password = user['PASSWORD']
                if(sha256_crypt.verify(password, user_password)):
                    session.permanent = True
                    session['id'] = user['ID']
                    session['email'] = user['EMAIL']
                    session['interest'] = user['INTEREST']
                    session['is_loggedin'] = True
                    return redirect(url_for('home'))
                else:
                    flash('Password is incorrect')
                    return render_template('auth/login.html')
            else:
                flash('Username is incorrect')
                return render_template('auth/login.html')
        else:
            return 'Please verify your account'
    else:
        return render_template('auth/login.html')

@app.route('/logout')
def logout():
    if 'email' in session:
        session.pop('id',None)
        session.pop('email',None)
        session.pop('is_loggedin',None)
        return redirect(url_for('login'))

@app.route('/register', methods=['POST', "GET"])
def register():
    if request.method == 'POST':
        username = request.form.get('username')
        first_name = request.form.get('first_name')
        last_name = request.form.get('last_name')
        email = request.form.get('email')
        phno = str(request.form.get('phone_number'))
        state = request.form.get('state')
        city = request.form.get('city')
        password = request.form.get('password')
        encrypt_password = sha256_crypt.encrypt(password)
        name = first_name + ' ' + last_name
        interest = request.form.get('interest')
        is_verified = False

        sql = "INSERT INTO
FVS49663.USER(username,name,email,phno,state,city,password,interest,is
_verified)
VALUES('{}','{}','{}','{}','{}','{}','{}','{}','{}')".format(username,name,email,ph
no,state,city,encrypt_password,interest,is_verified)
        ibm_db.exec_immediate(conn,sql)


        token = s.dumps(email, salt='email-confirm')
        msg = Message('Confirm Email', sender= os.getenv('FROM_EMAIL'),
recipients=[email])
        link = url_for('confirm_email', token=token + ',' +username,
_external=True)
        msg.body = 'Your link is {}'.format(link)
        mail.send(msg)
        return render_template('auth/verification_email_send.html')
    else:
        return render_template('auth/register.html')


@app.route('/confirm_email/<token>/')
def confirm_email(token):
    data = token.split(',')
    print(data)
    try:
        email = s.loads(data[0], salt='email-confirm', max_age=3600)

    except SignatureExpired:
        return '<h1>The token is expired!</h1>'
    sql = "UPDATE FVS49663.USER set is_verified = True where username =
'{}'".format(data[1])
    ibm_db.exec_immediate(conn,sql)
    return
render_template('auth/account_activation_success.html',is_success
=  True)




# Job routes
@app.route('/jobs',methods=['POST','GET'])
def jobs():
    if g.email:
        if request.method == 'POST':
            search_query = request.form.get('search_query')
            req =
requests.get("http://api.adzuna.com/v1/api/jobs/gb/search/1?app_id=b8
adcac4&app_key=68108197016f443e2c24af0587b39471&results_per_pag
e=50&what='{}'&content-type=application/json".format(search_query))
            jobs = json.loads(req.content)
            return render_template('jobs.html', jobs = jobs['results'])
        else:
            sql = "select * from FVS49663.USER where id = ?"
            stmt = ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt, 1, session['id'])
            ibm_db.execute(stmt)

            user = ibm_db.fetch_assoc(stmt)

            req =
requests.get("http://api.adzuna.com/v1/api/jobs/gb/search/1?app_id=b8
adcac4&app_key=68108197016f443e2c24af0587b39471&results_per_pag
e=50&what='{}'&content-
type=application/json".format(user['INTEREST']))
            jobs = json.loads(req.content)
            return render_template('jobs.html', jobs = jobs['results'])
    else:
        return redirect(url_for('login'))


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

```
def profile():
    if g.email:
        if request.method == 'POST':
            password = request.form.get('password')
            confirm_password = request.form.get('confirm_password')
            if password == confirm_password:
                print("Ok")
                user = session['id']
                encrypt_password = sha256_crypt.encrypt(password)
                sql = "UPDATE FVS49663.USER set password = '{}' where id =
'{}'".format(encrypt_password,user)
                ibm_db.exec_immediate(conn,sql)
                return redirect(url_for('profile'))
        else:
            sql = "select * from FVS49663.USER where id = ?"
            stmt = ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt, 1, session['id'])
            ibm_db.execute(stmt)

            user = ibm_db.fetch_assoc(stmt)

            sql1 = "select * from FVS49663.EDUCATION where id = ?"
            stmt1 = ibm_db.prepare(conn,sql1)
            ibm_db.bind_param(stmt1, 1, session['id'])
            ibm_db.execute(stmt1)

            educations = []
            dictionary = ibm_db.fetch_both(stmt1)
            while dictionary != False:
                educations.append(dictionary)
                dictionary = ibm_db.fetch_both(stmt1)

            print(educations)
            return
render_template('profile.html',user={"user":user,"educations":educations})

    else:
        return redirect(url_for('login'))

@app.route("/add_education",methods=['POST','GET'])
def add_education():
    if g.email:
        if request.method == 'POST':
            college_name = request.form.get('college_name')
            degree = request.form.get('degree')
            grade = request.form.get('grade')
            userid = session['id']
            sql = "insert into
FVS49663.EDUCATION(college_name,degree,grade,userid) values
('{}','{}','{}','{}')".format(college_name,degree,grade,userid)
            ibm_db.exec_immediate(conn,sql)
            return redirect(url_for('profile'))
        else:
            return render_template('add_education.html')
    else:
        return redirect(url_for('login'))


@app.route("/update_profile",methods=['POST','GET'])
def update_profile():
    if g.email:
        if request.method == 'POST':
            phno = str(request.form.get('phone_number'))
            state = request.form.get('state')
            city = request.form.get('city')
            interest = request.form.get('interest')
            userid = session['id']

            sql = "UPDATE FVS49663.USER set phno = '{}',state = '{}',city =
'{}',interest = '{}' where id = '{}'".format(phno,state,city,interest,userid)
            ibm_db.exec_immediate(conn,sql)
            return redirect(url_for('profile'))
        else:
            sql = "select * from FVS49663.USER where id = ?"
            stmt = ibm_db.prepare(conn,sql)
            ibm_db.bind_param(stmt, 1, session['id'])
            ibm_db.execute(stmt)

            user = ibm_db.fetch_assoc(stmt)
            return render_template('update_profile.html',user = user)
    else:
        return redirect(url_for('login'))
```

GITHUB & PROJECT DEMO LINK:

All the tasks of developing the application were uploaded on the github.
The github has been uploaded below.

https://github.com/IBM-EPBL/IBM-Project-46904-1660794589