



LOYOLA INSTITUTE OF TECHNOLOGY,CHENNAI

ANNA UNIVERSITY::CHENNAI – 600025

PROJECT TITLE: SKILL/JOB RECOMMENDER APPLICATION

A NALAIYA THIRAN PROJECT REPORT

SUBMITTED BY

TEAM SIZE: 4

TEAM LEADER: MUGESH M (210919205035)

TEAM MEMBER: MUGILVANNAN P (210919205036)

TEAM MEMBER: MOHAMMED AZHAR Z (210919205033)

TEAM MEMBER: JAGADISH P (210919205021)

TEAM ID: PNT2022TMID25734

MENTOR NAME: Mrs. VANITHA B

EVALUATOR NAME: Mrs.SAROJINI PREMALATHA J

TABLE OF CONTENT

CHAPTER	TITLE	PAGE NO
1.	INTRODUCTION	5
1.1	Project Overview	5
1.2	Purpose	6
2.	LITERATURE SURVEY	8
2.1	Existing Problem	9
2.2	References	8
2.3	Problem Statement Definition	9
3.	IDEATION & PROPOSED SOLUTION	11
3.1	Empathy Map Canvas	11
3.2	Ideation & Brainstorming	12
3.3	Proposed Solution	13
3.4	Problem Solution Fit	15
4.	REQUIREMENT ANALYSIS	16

4.1	Functional Requirement	16
4.2	Non-Functional Requirement	16
5.	PROJECT DESIGN	17
5.1	Data Flow Diagrams	18
5.2	Solution & Technical Architecture	19
5.3	User Stories	20
6.	PROJECT PLANNING & SCHEDULING	21
6.1	Sprint Planning & Estimation	21
6.2	Sprint Delivery & Schedule	22
6.3	Reports from JIRA	22
7.	CODING & SOLUTIONING	23
7.1	Feature 1	23
7.2	Feature 2	45
7.3	Database Schema	67
8.	TESTING	68

8.1	Test Cases	69
8.2	User Acceptance Testing	70
9.	RESULTS	73
9.1	Performance Metrics	74
10.	ADVANTAGES & DISADVANTAGES	92
11.	CONCLUSION	94
12.	FUTURE SCOPE	95
13.	APPENDIX	96
13.1	Source code	97
13.2	GitHub & Project Demo Link	101

1.INTRODUCTION

Nowadays, job search is a task commonly done on the Internet using job search engine sites like LinkedIn¹ , Indeed² , and others. Commonly, a job seeker has two ways to search a job using these sites:

1) doing a query based on keywords related to the job vacancy that he/she is looking for,

2) creating and/or updating a professional profile containing data related to his/her education, professional experience, professional skills and other, 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

1.1Project overview

This Project view provides an overview of the skill and job recommended for individuals interested in a career in any fields. It discusses the important role that any field plays in businesses and the various skills that are necessary for success in this field. It also outlines the different job opportunities available in any field and the different types of companies that employ any field professionals.

1.1 Purpose

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

To develop an end to end web application capable of displaying the current job openings based on the skillset of the users. The users and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. User will interact with the chatbot and can get the recommendations based on his skills. We can use jobsearch 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

References link:

1. https://www.researchgate.net/publication/272802616_A_survey_of_job_recommender_systems
2. https://www.researchgate.net/publication/360820692_Intelligent_Chatbot
3. Journal homepage: <http://www.ijrpr.com/> ISSN 2582-7421
4. <https://www.ijresm.com/>

2.3 Problem Statement Definition

Dealing with the enormous amount of recruiting information on the internet, a job seeker always spends hours to find useful ones. Many times, people who lack industry knowledge are unclear about what exactly they need to learn in order to get a suitable job for them. We address the problem of recommending suitable jobs to people who are seeking a new job.

Job recommender technology aims to help job seekers in finding jobs that match their skills.

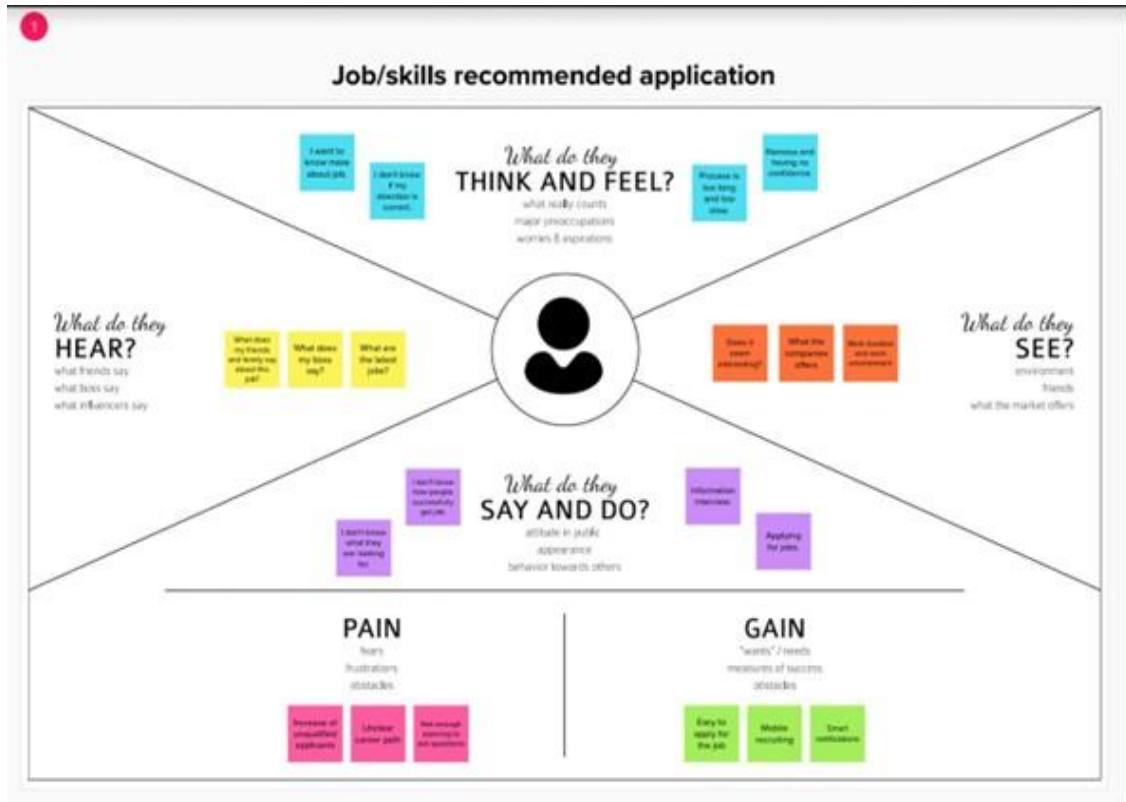
The internet caused a substantial impact on the recruitment process through the creation of e-recruiting platforms that become a primary recruitment channel in most companies. While companies established job positions on these portals, job-seeker uses them to publish their profiles. E-Recruitment platforms accomplished clear advantages for both recruiters and job-seekers by reducing the recruitment time and advertisement cost

.Recommender system technology aims to help users in finding items that match their preferences; it has a successful usage in a wide-range of applications to deal with problems related to information overload efficiently. In order to improve the e-recruiting functionality, many

recommender system approaches have been proposed. This paper will analyze e-recruiting process and related issues for building personalized recommender system of candidates.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation and Brain storming

The image displays a digital ideation workspace consisting of seven vertical panels and a bottom toolbar. The panels are as follows:

- Panel 1:** Features a light blue background with a pink vertical bar on the left. It contains a lightbulb icon, a list of small blue squares, and a small blue rectangle at the bottom.
- Panel 2:** Features a light blue background with a pink vertical bar on the left. It contains a small blue rectangle at the bottom.
- Panel 3:** Features a light blue background with a pink vertical bar on the left. It contains a small blue rectangle at the bottom.
- Panel 4:** Features a light blue background with a pink vertical bar on the left. It contains a small blue rectangle at the bottom.
- Panel 5:** Features a light blue background with a pink vertical bar on the left. It contains a small blue rectangle at the bottom.
- Panel 6:** Features a light blue background with a pink vertical bar on the left. It contains a small blue rectangle at the bottom.
- Panel 7:** Features a light blue background with a pink vertical bar on the left. It contains a small blue rectangle at the bottom.

The bottom toolbar is a dark grey bar containing several icons for editing and navigation, including a selection tool, a text tool, a shape tool, a line tool, a eraser, a lasso, a zoom in/out, a reset, and a save icon.

3.3 Proposed Solution

Team ID

PNT2022TMID25734

PROJECT NAME

SKILL & JOB RECOMMENDER

S.NO

	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	<p>Nowadays a lot of students have great skills but unable to get a desired/appropriate job, so an end-to-end web application can be created which is capable of displaying current job openings based on user skill set making it easier to hire and get hired.</p>
2.	Idea/solution description	<p>To develop an end-to-end web application which in default have a lot of current job openings through job search API out of which appropriate job will be recommended based on user skill set. At the same time students can develop their skills side by side with various courses and webinars offered by reputed organization. In addition to this a smart chat bot will be available for 24*7 which can help users in finding the right job.</p> <p>Though we have a lot of job searching applications, this one is unique because,</p> <ul style="list-style-type: none">❖ We have a smart chatbot built with IBM Watson

3. **Novelty/Uniqueness**

- ❖ Our platform not only helps in getting job but also helps in developing skills to get right job
- ❖ Here you can save/bookmark jobs for later use and also turn on notification for company specific job alerts
- ❖ Add media files to your profile to showcase your achievements
- ❖ It is made responsive to all screen sizes

4. **Social Impact/ Customer Satisfaction**

Students will be benefited as they will get to know which job suits them based on their skill set and therefore Lack of Unemployment can be reduced.

5. **Business Model (Revenue Model)**

6. **Scalability of the Solution**

We can provide the application for job seekers in a subscription based and we can share the profiles with companies and generate the revenue by providing them best profiles.

Data can be scaled up and scaled down according to number of current job openings available.

3.4 Problem Solution Fit

Define CS, fit into CC	1.CUSTOMER SEGMENTS 1) Jobless people 2) New college grads	6.CUSTOMER CONSTRAINTS For the website to operate as intended, basic needs such an internet connection and laptop are required.	5.AVAILABLE SOLUTIONS Earlier, job seekers used TV adverts and paper columns, as a result of the expanding digital world,the use of suggestion websites.	Explore AS,differentiate
focus on J&P, tap into BE.	2.JOBS-TO-BE-DONE/PROBLEM Make some work recommender site with an inbuilt chatbot help	9.PROBLEM ROOT CAUSE The vast majority don't know about their positions accessible in the market/sites	7.BEHAVIOURS The users attempt to first analyse job searches on websites, papers, and adverts depending on their requirements.	focus on J&P, tap into BE
Identify strong TR&EM	3.TRIGGERS Seeing other find a new line of work 4.EMOTIONS:BEFORE/AFTER User will be satisfied with the services and higher possibility of job offer	10.YOUR SOLUTION To build a platform that helps freshersand under graduates to get a job	8.CHANNELS OF BEHAVIOUR ONLINE : Ready to explore a suitable job based on their skill sets and necessities OFFLINE : Attend interviews on-siteand try and get a job	Identify strong TR&EM

1. REQUIREMENT ANALYSIS

1.1.Function Requirement

Software Required:

Python, Flask, Docker

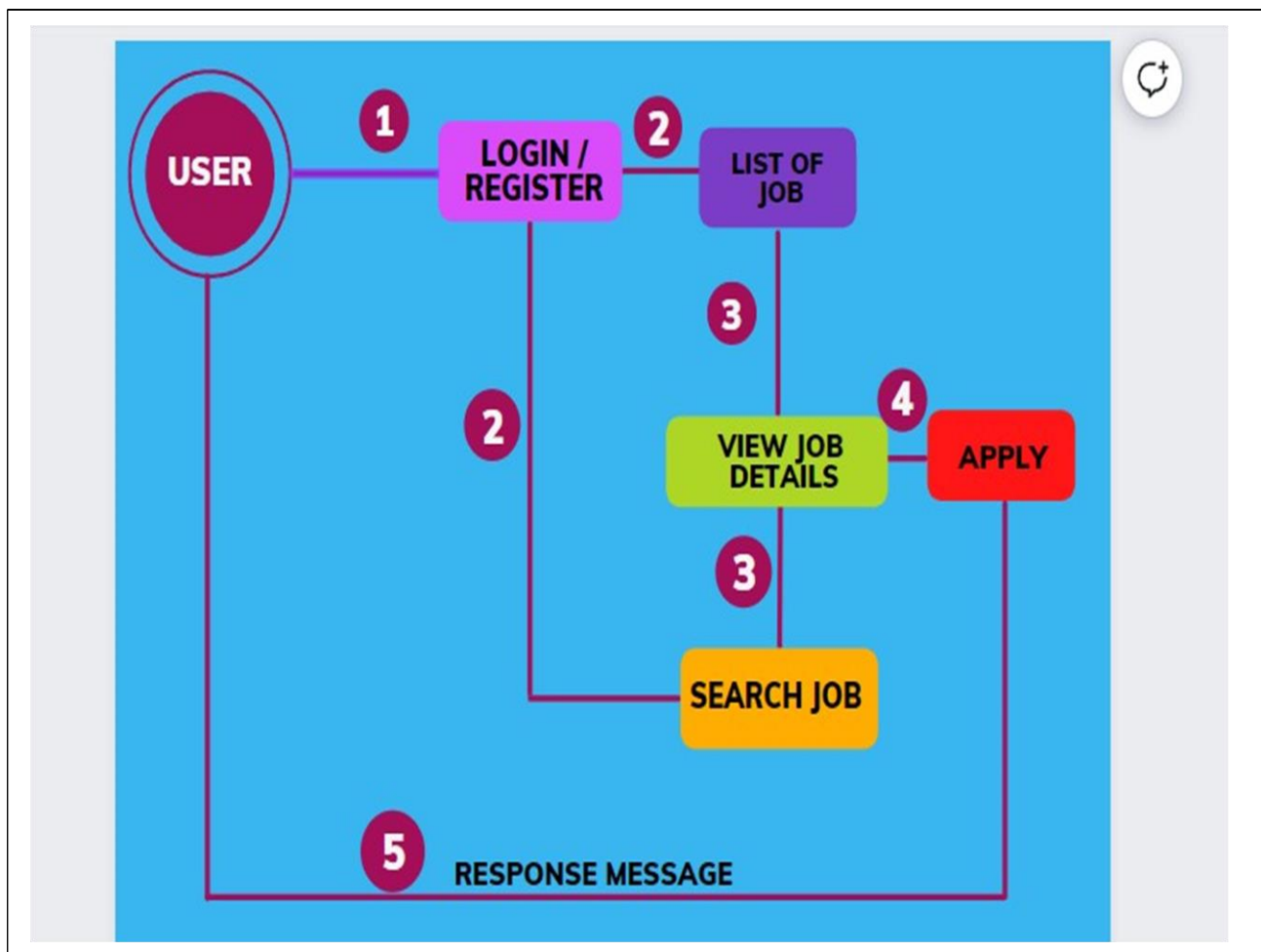
1.2. Non-Function Requirement

System Required:

8GB RAM, Intel Core i3, OS-Windows/Linux/MAC,
Laptop or Desktop

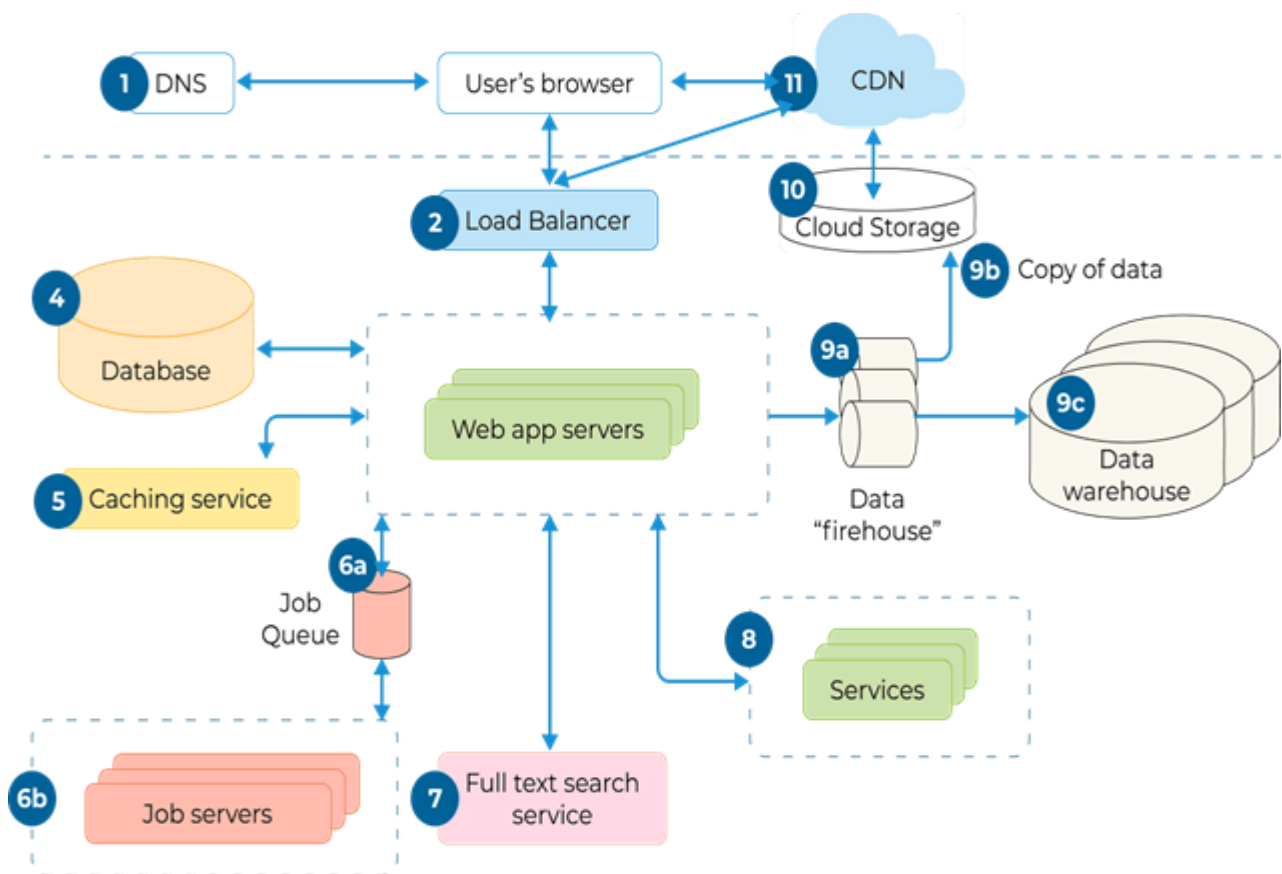
2. PROJECT DESIGN

2.1.Data Flow Diagrams

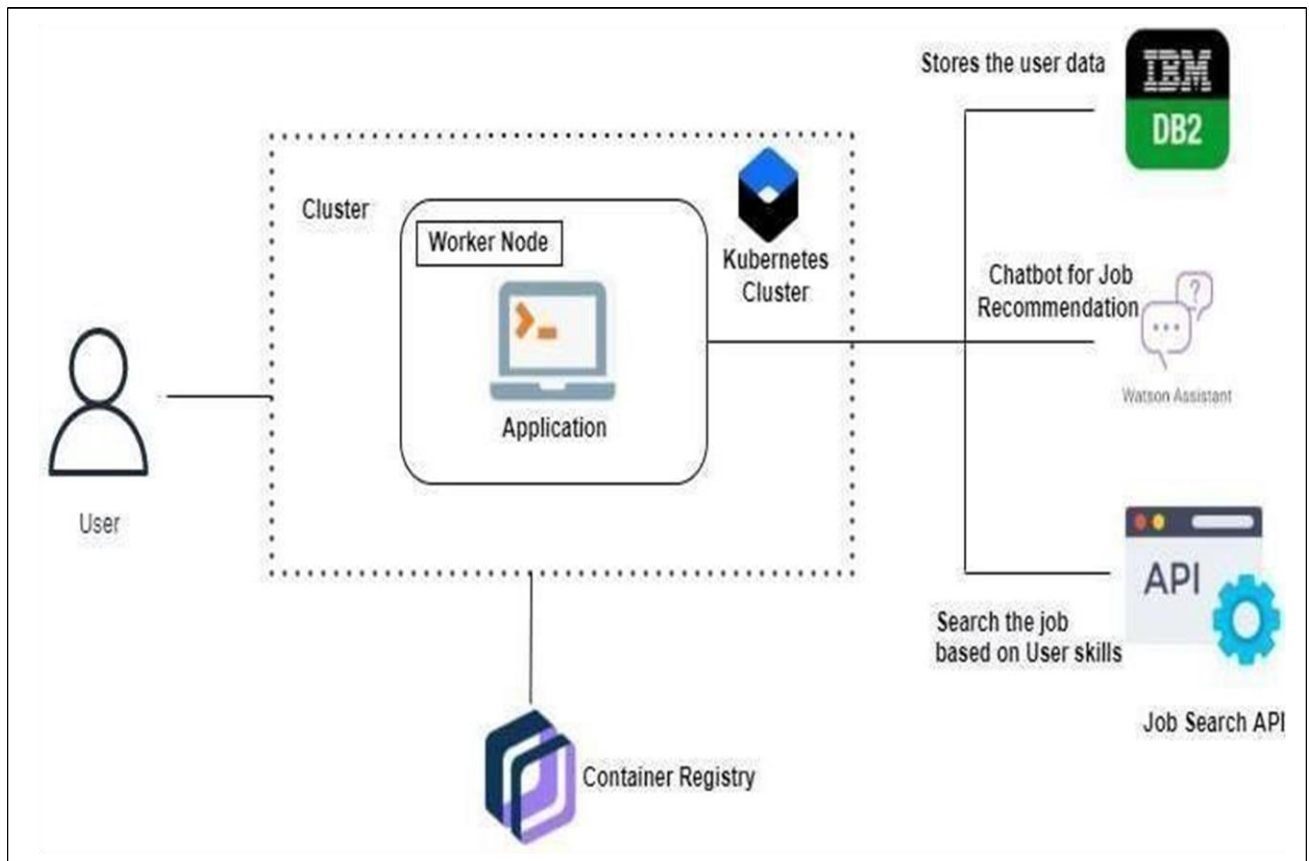


2.2. Solution & Technical Architecture

SOLUTION ARCHITECTURE:



TECHNICAL ARCHITECTURE:



2.3. User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	Create a model set that contains those models, then assign it to a role.	Assign that group to the appropriate roles on the Roles page	High	Sprint-1
Customer (Web user)	Identity-Aware	USN-7	Open, public access, User-authenticated access, Employee-restricted access.	Company public website. App running on the company intranet. App with access to customer private information.	High	Sprint-1
Customer Care Executive	Communication	USN-8	A customer care executive is a professional responsible for communicating the how's and why's regarding service expectations within a company.	For how to tackle customer queries.	Medium	Sprint-1
Administrator	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the directory.	Ease of use.	Medium	Sprint-1

3. PROJECT PLANNING & SCHEDULING

3.1. Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Mugesh, Mugilvannan
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Mugilvannan, Jagadish
Sprint-2		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Mohammed Azhar, Jagadish
Sprint-3		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Mugesh
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Mugesh
Sprint-2	Dashboard	USN-6	Create a model set that contains those models, then assign it to a role.	Assign that group to the appropriate roles on the Roles page	High	Mugilvannan, Mugesh
Sprint-4		USN-7	Open, public access, User-authenticated access, Employee-restricted access.	Company public website. App running on the company intranet. App with access to customer private information.	High	Mohammed Azhar
Sprint-1	Communication	USN-8	A customer care executive is a professional responsible for communicating the how's and why's regarding service expectations within a company.	For how to tackle customer queries.	Medium	Jagadish
Sprint-3	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the directory.	Ease of use.	Medium	Mugesh, Mohammed Azhar

3.2. Sprint Delivery Schedule

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	18	06 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	11 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	19	19 Nov 2022

VELOCITY:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

3.3. Report from JIRA

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

4. CODING & SOLUTIONING

4.1. Feature 1

JOB SEEKER

```
<!DOCTYPE html>

<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

  <style>
    .rus{

      border-radius: 0%
    };

    .black{
      color:black;
    }
  </style>

</head>
<body>

<!--=====-->
<nav class="navbar navbar-inverse rus">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Skill and Job Recommender</a>
    </div>

    <ul class="nav navbar-nav navbar-right">
      <li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Log
out</a></li>
    </ul>
```

```

    </div>
</nav>
<!--=====-->

<div class="container">
  <ul class="nav nav-tabs">
    <li class="active"><a data-toggle="tab" href="#home">Profile</a></li>
    <li><a data-toggle="tab" href="#menu1">Eligibility Test</a></li>
    <li><a data-toggle="tab" href="#menu2">Find Job</a></li>

  </ul>

  <div class="tab-content">
    <div id="home" class="tab-pane fade in active">
      <h3>PROFILE</h3>
      <hr>

      <form>
        <div class="alert alert-danger">
          <div class="form-group">
            <label for="name"><span class="black">Name: </span></label>
            <input type="text" class="form-control" id="name" value="Mahadevan">
          </div>
          <div class="form-group">
            <label for="email"><span class="black">Email Address </span></label>
            <input type="text" class="form-control" id="email"
value="Mahadevan@gmail.com">
          </div>

          <div class="form-group">
            <label for="email"><span class="black">Mobile No </span></label>
            <input type="text" class="form-control" id="email" value="9003824096">
          </div>

          <div class="form-group">
            <label for="quali"><span class="black">What is your Nationality
</span></label>
            <input type="text" class="form-control" id="quali" value="India">
          </div>

          <div class="form-group">
            <label for="quali">What is your Qualification</label>
            <input type="text" class="form-control" id="quali" value="BE">
          </div>

```

```

        <div class="form-group">
            <label for="quali">What year did you complete your bachelor
Degree?</label>
            <input type="text" class="form-control" id="quali" value="2023">
        </div>

        <div class="form-group">
            <label for="experience">How many years of work experience do you have
?</label>
            <input type="text" class="form-control" id="experience" value="0">
        </div>

        <div class="form-group">
            <label for="place">Where have you graduated high school degree ?</label>
            <input type="text" class="form-control" id="place" value="Government high
school Ariyalur">
        </div>

        <div class="form-group">
            <label for="SSLC">What is your SSLC Percentage :</label>
            <input type="text" class="form-control" id="SSLC" value="90">
        </div>

        <div class="form-group">
            <label for="HSC">What is your HSC Percentage :</label>
            <input type="text" class="form-control" id="HSC" value="90">
        </div>

        <div class="form-group">
            <label for="degree">What is your Percentage in bachelor Degree:</label>
            <input type="text" class="form-control" id="degree" value="90.5">
        </div>

        <textarea class="form-control" id="aboutyou" rows="3" placeholder="Write
few words about yourself"></textarea> <br> <br>

        <button type="submit" class="btn btn-primary btn-success">Save
changes</button> <br> <br>
    </div>
</form>
</div>

```



```

<div id="menu1" class="tab-pane fade">
  <h3>Eligibility Test </h3> <hr>

  <div class="alert alert-info">
    <label>What function is used to display output on the screen?</label>
    <div class="radio">
      <label><input type="radio" name="print">printf()</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="scanf">scanf()</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="switch">switch</label>
    </div>
  </div> <hr>

  <div class="alert alert-info">
    <label>What function is used to display output on the screen?</label>
    <div class="radio">
      <label><input type="radio" name="print">printf()</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="scanf">scanf()</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="switch">switch</label>
    </div>
  </div> <hr>

  <div class="alert alert-info">
    <label>What function is used to display output on the screen?</label>
    <div class="radio">
      <label><input type="radio" name="print">printf()</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="scanf">scanf()</label>
    </div>
    <div class="radio">
      <label><input type="radio" name="switch">switch</label>
    </div>
  </div> <hr>

```

```

<div class="alert alert-info">
  <label>What function is used to display output on the screen?</label>
  <div class="radio">
    <label><input type="radio" name="print">printf()</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="scanf">scanf()</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="switch">switch</label>
  </div>
</div> <hr>

<div class="alert alert-info">
  <label>What function is used to display output on the screen?</label>
  <div class="radio">
    <label><input type="radio" name="print">printf()</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="scanf">scanf()</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="switch">switch</label>
  </div>
</div><hr>

<div class="alert alert-info">
  <label>What function is used to display output on the screen?</label>
  <div class="radio">
    <label><input type="radio" name="print">printf()</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="scanf">scanf()</label>
  </div>
  <div class="radio">
    <label><input type="radio" name="switch">switch</label>
  </div>
</div>

  <button type="submit" class="btn btn-primary btn-success">Submit the
test</button> <br> <br>
</div>

<div id="menu2" class="tab-pane fade">
  <h3>Find the job</h3> <hr>
  <br>

```

```

<div class="container">
  <!-- ..... -->
  <form>
    <div class="input-group">
      <input type="text" class="form-control" id="myInput" placeholder="Look for a
job">
      <div class="input-group-btn">
        <button class="btn btn-default" type="submit">
          <i class="glyphicon glyphicon-search"></i>
        </button>
      </div>
    </div>
  </form> <br> <br>

  <!-- ..... -->

  <div style="width: 600px;">
    <!-- Left-aligned -->
    <div class="media">
      <div class="media-left">
        
      </div>
      <div class="media-body">
        <h4 class="media-heading">Java Developer</h4>
        <h5 class="media-heading"> Company Name : JK Software Solution</h5>
        <p>Job Location : Chennai</p>
        <p> Salary : 45000</p>
        <button type="submit" class="btn btn-primary btn-success">Apply </button>
      </div>
    </div>
    <br> <br>

    <div style="width: 600px;">
      <!-- Left-aligned -->
      <div class="media">
        <div class="media-left">
          
        </div>
        <div class="media-body">
          <h4 class="media-heading">.NET Developer</h4>
          <h5 class="media-heading"> Company Name : JK Software Solution</h5>

```

```

        <p>Job Location : Chennai</p>
        <p> Salary : 45000</p>
        <button type="submit" class="btn btn-primary btn-success">Apply </button>
<br> <hr>
    </div>
</div>
</div>

<div style="width: 600px;">
    <!-- Left-aligned -->
    <div class="media">
        <div class="media-left">
            
        </div>
        <div class="media-body">
            <h4 class="media-heading">PHP Developer</h4>
            <h5 class="media-heading"> Company Name : JK Software Solution</h5>
            <p>Job Location : Chennai</p>
            <p> Salary : 45000</p>
            <button type="submit" class="btn btn-primary btn-success">Apply </button>
<br> <hr>
        </div>
    </div>
</div>

<div style="width: 600px;">
    <!-- Left-aligned -->
    <div class="media">
        <div class="media-left">
            
        </div>
        <div class="media-body">
            <h4 class="media-heading">UX /UI Designer</h4>
            <h5 class="media-heading"> Company Name : JK Software Solution</h5>
            <p>Job Location : Chennai</p>
            <p> Salary : 45000</p>
            <button type="submit" class="btn btn-primary btn-success">Apply </button>
<br> <hr>
        </div>
    </div>
</div>

</div>

```

```
    </div>
</div>

<script>
    $(document).ready(function(){
        $("#myInput").on("keyup", function() {
            var value = $(this).val().toLowerCase();
            $("#myDIV *").filter(function() {
                $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)
            });
        });
    });
</script>

</body>

</html>
```

BACK END OF FEATURE 1

```
from flask import *#importing flask (Install it using python -m pip install flask)
import os
import ibm_db
import bcrypt
#import send_from_directory
app = Flask(__name__) #initialising flask
app.secret_key = ''
PEOPLE_FOLDER = os.path.join('static', 'people_photo')

import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45Raj9A0rr",'','')

@app.route("/") #defining the routes for the home() funtion (Multiple routes can be used as seen here)
@app.route("/home")

def dashboard():
    return render_template("dashboard.html") #rendering our home.html contained within /templates

def createjobseeker_profile():
    if request.method == 'POST':
        name = request.form['name']
        email = request.form['email']
        mobileno=request.form['mobileno']
        nationality=request.form['nationality']
        qualification=request.form['qualification']
        passout=request.form['passout']
        experience=request.form['experience'];
        degree=request.form['degree'];
        per_sslc=request.form['per_sslc'];
        per_hsc=request.form['per_hsc'];
        per_degree=request.form['per_degree'];

        tum = request.files['resume']
```

```

tum.save(os.path.join(app.config['UPLOAD_DIR'],tum.filename))

hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"
stmt = ibm_db.prepare(conn, query)
ibm_db.bind_param(stmt,1,email)
ibm_db.execute(stmt)
isUser = ibm_db.fetch_assoc(stmt)

if not isUser:
    insert_sql = "INSERT INTO JOBSEEKER (name, email, mobileno, nationality,
qualification, passout, experience, degree,per_sslc,per_hsc,per_degree) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, name)
    ibm_db.bind_param(prepare_stmt, 2, email)
    ibm_db.bind_param(prepare_stmt, 3, mobileno)
    ibm_db.bind_param(prepare_stmt, 4, nationality)
    ibm_db.bind_param(prepare_stmt, 5, qualification)
    ibm_db.bind_param(prepare_stmt, 6, passout)
    ibm_db.bind_param(prepare_stmt, 7, experience)
    ibm_db.bind_param(prepare_stmt, 8, degree)
    ibm_db.bind_param(prepare_stmt, 9, per_sslc)
    ibm_db.bind_param(prepare_stmt, 10, per_hsc)
    ibm_db.bind_param(prepare_stmt, 11, per_degree)
    ibm_db.execute(prepare_stmt)
    return render_template('dashboard.html')

return render_template('dashboard.html',name='Home')

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

```

4.2. Feature 2

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>Bootstrap Example</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-
scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst
rap.min.css">

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>

  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra
p.min.js"></script>

  <style>

    .rus{

      border-radius: 0%

    };

    .black{

      color:black;

    }

  </style>

</head>

<body>

<nav class="navbar navbar-inverse rus">

  <div class="container-fluid">

    <div class="navbar-header">
```



```

        <a class="navbar-brand" href="#">Skill and Job
Recommender</a>

    </div>

    <ul class="nav navbar-nav navbar-right">

        <li><a href="#"><span class="glyphicon glyphicon-log-
in"></span> Log out</a></li>

    </ul>

</div>

</nav>

<div class="container">

    <ul class="nav nav-tabs">

        <li class="active"><a data-toggle="tab" href="#home">Company
Profile</a></li>

        <li><a data-toggle="tab" href="#menu2">Find
candidate</a></li>

    </ul>

    <div class="tab-content">

        <div id="home" class="tab-pane fade in active">

            <h3>COMPANY PROFILE</h3>

            <hr>

            <form>

                <div class="alert alert-danger">

                    <div class="form-group">

                        <label for="name"><span class="black">Company Name:
</span></label>

                        <input type="text" class="form-control" id="name"
value="Mahadevan">

                    </div>

                    <div class="form-group">

```

```
        <label for="email"><span class="black">Company Email  
Address </span></label>
```

```
        <input type="text" class="form-control" id="email"  
value="Mahadevan@gmail.com">
```

```
    </div>
```

```
    <div class="form-group">
```

```
        <label for="email"><span class="black">Mobile No  
</span></label>
```

```
        <input type="text" class="form-control" id="email"  
value="9003824096">
```

```
    </div>
```

```
    <div class="form-group">
```

```
        <label for="quali"><span  
class="black">Location</span></label>
```

```
        <input type="text" class="form-control" id="quali"  
value="India">
```

```
    </div>
```

```
    <textarea class="form-control" id="aboutyou" rows="3"  
placeholder="Write few words about company"></textarea> <br>  
<br>
```

```
    <button type="submit" class="btn btn-primary btn-  
success">Save changes</button> <br> <br>
```

```
    </div>
```

```
    </form>
```

```
</div>
```

```
<div id="menu2" class="tab-pane fade">
```

```
    <h3>Find the Condidate</h3> <hr>
```

```
    <br>
```

```

    <div class="container">
    <!--.....
>
    <form>
    <div class="input-group">
        <input type="text" class="form-control" id="myInput"
placeholder="Look for a candidate">
    <div class="input-group-btn">
        <button class="btn btn-default"
type="submit">
            <i class="glyphicon glyphicon-
search"></i>
        </button>
    </div>
    </div>
    </form> <br> <br>
    <!--.....>
    <div style="width: 600px;">
        <!-- Left-aligned -->
        <div class="media">
            <div class="media-left">
                
            </div>
            <div class="media-body">
                <h4 class="media-heading">Name :
Mahadevan</h4>
                <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>

```

```

        <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>

        </div>

    </div>

</div>

<br>
<div style="width: 600px;">
    <!-- Left-aligned -->
    <div class="media">
        <div class="media-left">
            
        </div>
        <div class="media-body">
            <h4 class="media-heading">Name :
Manikandan</h4>
            <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>
            <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>
        </div>
    </div>
</div> <br>
<div style="width: 600px;">
    <!-- Left-aligned -->
    <div class="media">
        <div class="media-left">
            
        </div>

```

```

        <div class="media-body">
            <h4 class="media-heading">Name :
Kavi</h4>
            <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>
            <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>
        </div>
    </div>
</div> <br>

<div style="width: 600px;">
    <!-- Left-aligned -->
    <div class="media">
        <div class="media-left">
            
        </div>
        <div class="media-body">
            <h4 class="media-heading">Name :
Ram</h4>
            <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>
            <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>
        </div>
    </div>
</div> <br>
</div>
</div>

```

```

</div>

<script>
    $(document).ready(function() {
        $("#myInput").on("keyup", function() {
            var value = $(this).val().toLowerCase();
            $("#myDIV *").filter(function() {
                $(this).toggle($(this).text().toLowerCase().indexOf(value)
> -1)
            });
        });
    });
</script>
</body>
</html>

```

CHATBOT :

Chatbot has been implemented to provide assistance .

```

window.watsonAssistantC
hatOptions = {
    integrationID: "d73273d3-3f44-4304-
84ee-8fd243016d1d", // The ID of this
integration.
    region: "jp-tok", // The region
your integration is hosted in.
    serviceInstanceID: "81229104-ee6b-46da-
ac1c-67ede110663a", // 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.clie
ntVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });

```

4.3. Database Schema(if Applicable) :

```

import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail

# from_address we pass to our Mail object, edit with your name
FROM_EMAIL = 'Your_Name@SendGridTest.com'

def SendEmail(to_email):
    """ Send an email to the provided email addresses
    :param to_email = email to be sent to
    :returns API response code
    :raises Exception e: raises an exception """
    message = Mail(
        from_email=FROM_EMAIL,
        to_emails=to_email,
        subject='A Test from SendGrid!',
        html_content='<strong>Hello there from SendGrid your URL is: ' +
        '<a href='\"https://github.com/cyberjive\">right here!</a></strong>')
    try:
        sg = SendGridAPIClient(os.environ.get('SENDGRID_API_KEY'))
        response = sg.send(message)
        code, body, headers = response.status_code, response.body, response.headers
        print(f"Response Code: {code} ")
        print(f"Response Body: {body} ")
        print(f"Response Headers: {headers} ")
        print("Message Sent!")
    except Exception as e:
        print("Error: {0}".format(e))
    return str(response.status_code)

```

```
if _name_ == "_main_":
    SendEmail(to_email=input("Email address to send to? "))
```

5. TESTING:

5.1. Test Cases:

Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Functional	Login page	Verify that after registration users are navigated to login page	Mail id, Username, Password, Phone number, Pin	1. Open the website and go to register page. 2. Enter details and press register 3. Verify that users are navigated to registration page	https://drive.google.com/drive/folders/1Olfuud8BVX7mQv-JeYDMq41NmY1KF5S	Users should be navigated to registration page	Working as expected	Pass	Excellent	N		MUGILVANNAN
UI	Home Page	Verify the UI elements in Login/Signup popup	Username & Password	1. Open the website 2. Enter details and press login 3. Verify that users are notified of login process	https://drive.google.com/drive/folders/1OKpV13F1nZ1O6Sw2z9AIK8ELGsY9hRok	Users should be notified of login process	Not working	Fail	Trying To Recover	N	BUG-12	MOHAMMED AZHAR
Functional	Home page	Verify user is able to log into application with Valid credentials		1. Open the website 2. Enter details and press login 3. Verify that users are logged into website properly	Username: gr03122001@gmail.com password: 123	User should be logged into website properly	Working as expected	Pass	Good	N		MUGESH
Functional	Home Page	Verify that categories of skills and jobs are shown in homepage		1. Open the website 2. Enter details and press login 3. Verify that categories of are showing Jobs shown in homepage		Categories of skills and jobs should be shown in homepage	Working as expected	Pass	Good	N	BUG-14	JAGADISH
Functional	Home page	Verify that jobs are displayed in homepage		1. Open the website 2. Enter details and press login 3. Verify that jobs are displayed in homepage		Jobs should be displayed in homepage	Working as expected	Pass	Good	N		MUGILVANNAN
Functional	Home page	Verify that when clicked on jobs it is redirected to correct page		1. Open the website 2. Enter details and press login 3. Verify that when clicked on jobs it is redirected to correct page		When clicked on job link it should be redirected to correct page	Working as expected	Pass	Excellent	N		MUGESH

User Acceptance Testing:

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	1	3	1	6
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	80

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	1	7
Client Application	51	0	1	51
Security	2	0	2	2

6. RESULTS:

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.

7. 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.

8. 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 watson. The chatbot helps jobseeker and organization when they experience the difficulties.

9. 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.

10. APPENDIX

SOURCE CODE

Sprint 1:

Index.html

```
<!DOCTYPEhtml>

<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width, initial-scale=1">
    <title>Log-in Page </title>
    <link href="css/bootstrap.min.css"
rel="stylesheet">
    <script src="chat.js"></script>
  </head>
  <style>
    .pad{
      padding-top: 40%;
    }
    .bold{
      font-weight: bolder;
    }
    .backgroundcolour{
      background-color: beige;
    }
    .font{
      font-weight: bolder;
      color:whitesmoke;
    }
  </style>
  <body>

    <div class="container">
      <div class="row">
        <div class="col-sm-6">

          </div>
        <div class="col-sm-6">
```

```

<div class="pad ">

                                <center> </center> <hr>
                                <form>
                                    <div
class="input-group">
                                <span
class="input-group-addon"><i class="glyphicon
glyphicon-user"></i></span>
                                <input
id="email" type="text" class="form-control"
name="email" placeholder="Enter Your Registered
Email ID" required>
                                </div>
                                <br>
                                <div
class="input-group">
                                <span
class="input-group-addon"><i class="glyphicon
glyphicon-lock"></i></span>
                                <input
id="password" type="password" class="form-
control" name="password" placeholder="Enter Your
Password" required>
                                </div>

                                <hr>

<div class="row">
<div class="col-sm-6">
<button type="submit" id="" class="btn btn-
success btn-block"> <span class="font">LOG IN
</span></button>

                                </div>

                                <div class="col-sm-6">

                                <button id="" class="btn btn-danger btn-
block"><a href="forget_password.html"><span
class="font">FORGET PASSWORD

```

```

</span></a></button>

    </div>

<br>

    </div>

    </form>
    <a
href="signup.html"> <button type="submit" id=""
class="btn btn-primary btn-block"><span
class="font">CREATE AN ACCOUNT </span> </button>
</a>

    </div>

</div>

</div>
</div>

<script
src="js/jquery.min.js"></script>
<script
src="js/bootstrap.min.js"></script>
<script src="chat.js"></script>
</body>
</html>

```

FORGET-PASSWORD.html

```

<!DOCTYPEhtml>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width, initial-scale=1">
    <title>Responsive Sample</title>
    <link href="css/bootstrap.min.css"
rel="stylesheet">
  </head>
  <style>
    .pad{
      padding-top: 20%;
      padding-left: 20%;
    }
  </style>

```

```

        padding-right: 20%;
    }
    .bold{
        font-weight: bolder;
    }
</style>
<body>
    <div class="container">

        <div class="pad">
            <form>
                <center> <H3
class="bold">FORGET PASSWORD</H3></center> <hr>
                <div class="form-group">
                    <label for="email">Email
address:</label>
                    <input type="email"
class="form-control" id="email"
placeholder="Enter Your Registered Email
Address">
                </div>
                <div class="form-group">
                    <label
for="pwd">OTP</label>
                    <input type="password"
class="form-control" id="pwd" placeholder="Enter
your OTP ">
                </div>

                <button type="submit"
class="btn btn-default">Recover
Password</button>
            </form>
        </div>
    </div>

    <script
src="js/jquery.min.js"></script>
    <script
src="js/bootstrap.min.js"></script>
    </body>
</html>

```


SIGN UP.html

```
<!DOCTYPE
html>

<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport"
content="width=device-width,
initial-scale=1">
    <title>Creating New Account
</title>
    <link
href="css/bootstrap.min.css"
rel="stylesheet">
  </head>
  <style>
    .pad{
      padding-top: 10%;
      padding-bottom: 10%;
    }
    .bold{
      font-weight: bolder;
    }
  </style>
  <body>

    <div class="container">

      <div
class="pad">
        <h3
class="bold">CREATE AN ACCOUNT</h3>
        <HR>

        <form>

          <div
class="form-group">

            <label for="name">First
Name</label>

            <input type="name"
class="form-control" id="name"
placeholder="Enter Your First
Name">
```

```

        </div>
        <div
class="form-group">

        <label for="name">Last
Name</label>

        <input type="name"
class="form-control" id="name"
placeholder="Enter Your Last Name">

        </div>

        <div class="form-group">

        <label for="email">Email
address:</label>

        <input type="email"
class="form-control" id="email"
placeholder="Enter Your Email ID">

        </div>

        <div class="form-group">

        <label
for="pwd">Password:</label>

        <input type="password"
class="form-control" id="pwd"
placeholder="Enter Your Password">

        </div>
        <div
class="form-group">

        <label for="repwd">Reenter
Password:</label>

        <input type="password"
class="form-control" id="repwd"
placeholder="Enter Your Password
Again">

```

```
</div>
```

```
    <button type="button" id=""  
class="btn btn-primary btn-block  
btn-danger">CREATE AN  
ACCOUNT</button>
```

```
<hr>
```

```
    <button type="submit"  
id="login_already" class="btn btn-  
primary btn-block btn-success">Do  
you have an Account Already  
?</button>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
    <script  
src="js/jquery.min.js"></script>  
    <script  
src="js/bootstrap.min.js"></script>  
    <script  
src="chat.js"></script>  
    </body>  
</html>
```

BACK END :

```
import os
import ibm_db
import bcrypt
#import send_from_directory
app = Flask(__name__) #initialising flask
app.secret_key = ''
PEOPLE_FOLDER = os.path.join('static', 'people_photo')
import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45Raj9A0rr",'','')
@app.route("/") #defining the routes for the home() funtion
(Multiple routes can be used as seen here)
@app.route("/home")
def home():
    return render_template("home.html") #rendering our
home.html contained within /templates
@app.route("/login")
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        if not email or not password:
            return render_template('login.html',error='Please
fill all fields')

        query = "SELECT * FROM USER WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser,password)
        if not isUser:
            return
render_template('login.html',error='Invalid Credentials')
```

```

        isPasswordMatch =
bcrypt.checkpw(password.encode('utf-
8'),isUser['PASSWORD'].encode('utf-8'))
        if not isPasswordMatch:
            return
render_template('login.html',error='Invalid  Credentials')
        session['email'] = isUser['EMAIL']
        return redirect(url_for('user_dashboard'))
        return render_template('login.html',name='Home')
app.config["UPLOAD_DIR"] = "uploads"
@app.route("/register",methods=['GET','POST'])
def register():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        firstname=request.form['firstname']
        middlename=request.form['middlename']
        lastname=request.form['lastname']
        dob=request.form['dob']
        age=request.form['age']
        drno=request.form['drno']
        landmark1=request.form['lnmk1']
        landmark2=request.form['lnmk2']
        village =request.form['village']
        district=request.form['district']
        pincode=request.form['pincode']
        country=request.form['country']
        country_code=request.form['countryCode']
        phone=request.form['phone']
        schlname10=request.form['schlname10']
        maxmark10=request.form['maxmark10']
        actmark10=request.form['actmark10']
        schlname212=request.form['schlname212']
        maxmark212=request.form['maxmark212']
        actmark212=request.form['actmark212']
        clgname=request.form['clgname']
        clgmaxmark=request.form['clgmaxmark']
        clgactmark=request.form['clgactmark']
        stream=request.form['stream']
        graduation=request.form['graduation']
        grayear=request.form['grayear']
        branch=request.form['branch']
        skills=request.form['form[]']
        q1=request.form['Q1']
        q2=request.form['Q2']

```

```

q3=request.form['Q3']
q4=request.form['Q4']
q5=request.form['Q5']
position=request.form['position']
experience=request.form['expyears']
designation=request.form['designation']
tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD_DIR'],tum.filename))

hash=bcrypt.hashpw(password.encode('utf-
8'),bcrypt.gensalt())
query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"
stmt = ibm_db.prepare(conn, query)
ibm_db.bind_param(stmt,1,email)
ibm_db.execute(stmt)
isUser = ibm_db.fetch_assoc(stmt)

if not is User:
    insert_sql = "INSERT INTO JOBSEEKER (EMAIL, PASSWORD,
FIRSTNAME, MIDDLENAME, LASTNAME, DOB, AGE, DOOR_NO,
LANDMARK1, LANDMARK2,VILLAGE, DISTRICT, PINCODE, COUNTRY,
COUNTRY_CODE, PHONE, SCHOOL_NAME, SC_MAX_MARK,
SC_ACT_MARK,HSC_SCHOOL_NAME,HSC_MAX_MARK,
HSC_ACT_MARK,COLLEGE_NAME,COLLEGE_MAX,COLLEGE_ACT, STREAM,
GRADUATION, GRADUATION_YEAR, BRANCH, SKILLS, PRESENTATION,
HACKATHON, CODING, WON, PRIZE_WON, POSITION,
EXPYEARS,DESIGNATION, RESUME) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,
?,?,?,?,?,?,?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, email)
    ibm_db.bind_param(prepare_stmt, 2, password)
    ibm_db.bind_param(prepare_stmt, 3, firstname)
    ibm_db.bind_param(prepare_stmt, 4, middlename)
    ibm_db.bind_param(prepare_stmt, 5, lastname)
    ibm_db.bind_param(prepare_stmt, 6, dob)
    ibm_db.bind_param(prepare_stmt, 7, age)
    ibm_db.bind_param(prepare_stmt, 8, drno)
    ibm_db.bind_param(prepare_stmt, 9, landmark1)
    ibm_db.bind_param(prepare_stmt, 10, landmark2)
    ibm_db.bind_param(prepare_stmt, 11, village)
    ibm_db.bind_param(prepare_stmt, 12, district)
    ibm_db.bind_param(prepare_stmt, 13, pincode)
    ibm_db.bind_param(prepare_stmt, 14, country)

```

```

        ibm_db.bind_param(prepare_stmt, 15, country_code)
        ibm_db.bind_param(prepare_stmt, 16, phone)
        ibm_db.bind_param(prepare_stmt, 17, schlname10)
        ibm_db.bind_param(prepare_stmt, 18, maxmark10)
        ibm_db.bind_param(prepare_stmt, 19, actmark10)
        ibm_db.bind_param(prepare_stmt, 20, schlname212)
        ibm_db.bind_param(prepare_stmt, 21, maxmark212)
        ibm_db.bind_param(prepare_stmt, 22, actmark212)
        ibm_db.bind_param(prepare_stmt, 23, clgname)
        ibm_db.bind_param(prepare_stmt, 24, clgmaxmark)
        ibm_db.bind_param(prepare_stmt, 25, clgactmark)
        ibm_db.bind_param(prepare_stmt, 26, stream)
        ibm_db.bind_param(prepare_stmt, 27, graduation)
        ibm_db.bind_param(prepare_stmt, 28, grayear)
        ibm_db.bind_param(prepare_stmt, 29, branch)
        ibm_db.bind_param(prepare_stmt, 30, skills)
        ibm_db.bind_param(prepare_stmt, 31, q1)
        ibm_db.bind_param(prepare_stmt, 32, q2)
        ibm_db.bind_param(prepare_stmt, 33, q3)
        ibm_db.bind_param(prepare_stmt, 34, q4)
        ibm_db.bind_param(prepare_stmt, 35, q5)
        ibm_db.bind_param(prepare_stmt, 36, position)
        ibm_db.bind_param(prepare_stmt, 37, experience)
        ibm_db.bind_param(prepare_stmt, 38, designation)
        ibm_db.bind_param(prepare_stmt, 39, tum)
        ibm_db.execute(prepare_stmt)
        return render_template('register.html')
    else:
        return render_template('register.html',error='Invalid
Credentials')
    return render_template('register.html',name='Home')
@app.route("/orgregister")
def orgregister():
    return render_template("orgregister.html") #rendering our
register.html contained within /templates
@app.route("/user_dashboard")
def user_dashboard():
    return render_template("user_dashboard.html")
@app.route('/logout')
def logout():
    session.pop('email', None)
    return redirect(url_for('login'))
if __name__ == "__main__": #checking if __name__'s value is
'__main__'. __name__ is an python environment variable who's
value will always be '__main__' till this is the first

```

```
instatnce of app.py running
    app.run(debug=True,port=8080) #running flask (Initalised
on line 4)
```

SPRINT 2

Index.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <title>Log-in Page </title>

    <link href="css/bootstrap.min.css" rel="stylesheet">

    <script src="chat.js"></script>

  </head>

  <style>

    .pad{

      padding-top: 40%;

    }

    .bold{

      font-weight: bolder;
```



```
}

.backgroundcolour{
    background-color: beige;
}

.font{

    font-weight: bolder;
    color:whitesmoke;
}

</style>

<body>
    <br>

    <div class="container">

        <div class="row">
            <div class="col-sm-6">

                <div class="pad ">
```

```

<center> </center> <BR>

<form>

    <div class="input-group">

        <span class="input-group-
addon"><i class="glyphicon glyphicon-user"></i></span>

        <input id="email"
type="text" class="form-control" name="email" placeholder="Enter
Your Organization Registered Email ID" >

    </div>

    <br>

    <div class="input-group">

        <span class="input-group-
addon"><i class="glyphicon glyphicon-lock"></i></span>

        <input id="password"
type="password" class="form-control" name="password"
placeholder="Enter Your Password">

    </div>

    <hr>

    <div class="row">

        <div class="col-sm-6">

            <button type="submit" id="" class="btn btn-success btn-
block"> <span class="font">LOG IN </span></button>

        </div>

```

```
<div class="col-sm-6">

    <button id="" class="btn btn-danger btn-block">
<a href=""><span class="font">FORGET PASSWORD
</span></a></button>

</div>

</div> <br>

</form>

<a href="Organization/Create_Account/org_create_account.html">
<button type="submit" id="" class="btn btn-primary btn-
block"><span class="font">CREATE AN ACCOUNT </span> </button>
</a>

</div>

</div>

<div class="col-sm-6">

    <div class="pad ">
```

```

                                <center> </center> <br>

                                <form>

                                <div class="input-
group">

                                <span class="input-
group-addon"><i class="glyphicon glyphicon-user"></i></span>

                                <input id="email"
type="text" class="form-control" name="email" placeholder="Enter
Your Registered Email ID" required>

                                </div>

                                <br>

                                <div class="input-
group">

                                <span class="input-
group-addon"><i class="glyphicon glyphicon-lock"></i></span>

                                <input id="password"
type="password" class="form-control" name="password"
placeholder="Enter Your Password" required>

                                </div>

                                <hr>

                                <div class="row">

                                <div class="col-sm-
6">

```

```

                                <button
type="submit" id="" class="btn btn-success btn-block"> <span
class="font">LOG IN </span></button>

                                </div>

                                <div class="col-sm-
6">

                                <button id="" class="btn btn-danger btn-block">
<a href="forget_password.html">
<span class="font">FORGET PASSWORD </span></a></button>

                                </div>

                                </div> <br>

                                </form>

                                <a href="signup.html">
<button type="submit" id="" class="btn btn-primary btn-block">
<span class="font">CREATE AN ACCOUNT </span> </button>
</a></div>
</div>
</div>
</div>

    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="chat.js"></script>

</body>
</html>

```

FORGET-PASSWORD.html

```
<!DOCTYPEhtml>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport"
content="width=device-width, initial-scale=1">
  <title>Responsive Sample</title>
  <link href="css/bootstrap.min.css"
rel="stylesheet">
</head>
<style>
  .pad{
    padding-top: 20%;
    padding-left: 20%;
    padding-right: 20%;
  }
  .bold{
    font-weight: bolder;
  }
</style>
<body>
  <div class="container">

    <div class="pad">
      <form>
        <center> <H3
class="bold">FORGET PASSWORD</H3></center> <hr>
        <div class="form-group">
          <label for="email">Email
address:</label>
          <input type="email"
class="form-control" id="email"
placeholder="Enter Your Registered Email
Address">
        </div>
        <div class="form-group">
          <label
for="pwd">OTP</label>
          <input type="password"
class="form-control" id="pwd" placeholder="Enter
your OTP ">
      </form>
    </div>
  </div>
</body>
</html>
```

```
        </div>

        <button type="submit"
class="btn btn-default">Recover
Password</button>
    </form>
</div>
</div>

<script
src="js/jquery.min.js"></script>
<script
src="js/bootstrap.min.js"></script>
</body>
</html>
```

HOME PAGE:

Forgot password.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8">

        <meta name="viewport" content="width=device-width,
initial-scale=1">

    <title>Responsive Sample</title>

        <link href="css/bootstrap.min.css"
rel="stylesheet">

  </head>

  <style>

      .pad{

        padding-top: 20%;

        padding-left: 20%;

        padding-right: 20%;

      }

      .bold{

        font-weight: bolder;

      }

  </style>

  <body>

    <div class="container">

      <div class="pad">
```



```

        <form>

            <center> <H3 class="bold">FORGET
PASSWORD</H3></center> <hr>

            <div class="form-group">

                <label for="email">Email
address:</label>

                <input type="email" class="form-
control" id="email" placeholder="Enter Your Registered Email
Addtress">

            </div>

            <div class="form-group">

                <label for="pwd">OTP</label>

                <input type="password" class="form-
control" id="pwd" placeholder="Enter your OTP ">

            </div>

            <button type="submit" class="btn btn-
default">Recover Password</button>

        </form>

    </div>

    </div>

    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</body>

</html>

```

Signup.html

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width,
initial-scale=1">

    <title>Creating New Account </title>

    <link href="css/bootstrap.min.css"
rel="stylesheet">

  </head>

  <style>

    .pad{

      padding-top: 10%;

      padding-bottom: 10%;

    }

    .bold{

      font-weight: bolder;

    }

  </style>

  <body>

    <div class="container">
```

```

        <div class="pad">
            <h3 class="bold">CREATE AN
ACCOUNT</h3>

            <HR>

            <form>

                <div class="form-group">

                    <label
for="name">First Name</label>

                    <input type="name"
class="form-control" id="name" placeholder="Enter Your First
Name">

                </div>

                <div class="form-group">

                    <label
for="name">Last Name</label>

                    <input type="name"
class="form-control" id="name" placeholder="Enter Your Last
Name">

                </div>

                <div class="form-group">

```

```

                                <label
for="email">Email address:</label>

                                <input type="email"
class="form-control" id="email" placeholder="Enter Your Email
ID">

                                </div>

                                <div class="form-group">

                                    <label
for="pwd">Password:</label>

                                    <input
type="password" class="form-control" id="pwd" placeholder="Enter
Your Password">

                                    </div>

                                <div class="form-group">

                                    <label
for="repwd">Reenter Password:</label>

                                    <input
type="password" class="form-control" id="repwd"
placeholder="Enter Your Password Again">

                                    </div>

                                <button
type="button" id="" class="btn btn-primary btn-block btn-
danger">CREATE AN ACCOUNT</button>

                                <hr>

```

```
                                <button  
type="submit" id="login_already" class="btn btn-primary btn-  
block btn-success">Do you have an Account Already ?</button>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
</div>
```

```
<script src="js/jquery.min.js"></script>
```

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

```
<script src="chat.js"></script>
```

```
</body>
```

```
</html>
```

BACK END

APP.PY

```
from flask import *#importing flask (Install it using python -m
pip install flask)

import os

import ibm_db

import bcrypt

#import send_from_directory

app = Flask(__name__) #initialising flask

app.secret_key = ''

PEOPLE_FOLDER = os.path.join('static', 'people_photo')


import ibm_dbconn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-
7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT
=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.cr
t;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45Raj9A0rr",'','')


@app.route("/") #defining the routes for the home() funtion
(Multiple routes can be used as seen here)

@app.route("/home")

def home():

    return render_template("index/INDEX.html") #rendering our
home.html contained within /templates
```

```

@app.route("/login")
def login():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        if not email or not password:
            return render_template('login.html',error='Please
fill all fields')

        query = "SELECT * FROM USER WHERE email=?"
        stmt = ibm_db.prepare(conn, query)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        isUser = ibm_db.fetch_assoc(stmt)
        print(isUser,password)

        if not isUser:
            return render_template('login.html',error='Invalid
Credentials')

        isPasswordMatch = bcrypt.checkpw(password.encode('utf-
8'),isUser['PASSWORD'].encode('utf-8'))

        if not isPasswordMatch:
            return render_template('login.html',error='Invalid
Credentials')

```

```

        session['email'] = isUser['EMAIL']

        return redirect(url_for('user_dashboard'))

    return render_template('login.html',name='Home')

app.config["UPLOAD_DIR"] = "uploads"

@app.route("/register",methods=['GET','POST'])
def register():
    if request.method == 'POST':
        email = request.form['email']
        password = request.form['password']
        firstname=request.form['firstname']
        middlename=request.form['middlename']
        lastname=request.form['lastname']
        dob=request.form['dob']
        age=request.form['age']
        drno=request.form['drno']
        landmark1=request.form['lnmk1']
        landmark2=request.form['lnmk2']
        village =request.form['village']
        district=request.form['district']
        pincode=request.form['pincode']

```



```
country=request.form['country']
country_code=request.form['countryCode']
phone=request.form['phone']
schlname10=request.form['schlname10']
maxmark10=request.form['maxmark10']
actmark10=request.form['actmark10']
schlname212=request.form['schlname212']
maxmark212=request.form['maxmark212']
actmark212=request.form['actmark212']
clgname=request.form['clgname']
clgmaxmark=request.form['clgmaxmark']
clgactmark=request.form['clgactmark']
stream=request.form['stream']
graduation=request.form['graduation']
grayear=request.form['grayear']
branch=request.form['branch']
skills=request.form['form[]']
q1=request.form['Q1']
q2=request.form['Q2']
q3=request.form['Q3']
q4=request.form['Q4']
q5=request.form['Q5']
position=request.form['position']
experience=request.form['expyears']
designation=request.form['designation']
```

```

tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD_DIR'],tum.filename))


hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())


query = "SELECT EMAIL FROM ORGANIZATION WHERE EMAIL=?"
stmt = ibm_db.prepare(conn, query)
ibm_db.bind_param(stmt,1,email)
ibm_db.execute(stmt)
isUser = ibm_db.fetch_assoc(stmt)

if not isUser:

    insert_sql = "INSERT INTO ORGANIZATION (ORGANIZATION_NAME,
ORGANIZATION_ID, ORGANIZATION_ADDRESS,SET_PASSWORD,
REENTERPASSWORD) VALUES (?, ?, ?, ?, ?)"

    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prepare_stmt, 1, ORGANIZATION_NAME)
    ibm_db.bind_param(prepare_stmt, 2, ORGANIZATION_ID)
    ibm_db.bind_param(prepare_stmt, 3, ORGANIZATION_ADDRESS)
    ibm_db.bind_param(prepare_stmt, 4, SET_PASSWORD)
    ibm_db.bind_param(prepare_stmt, 5, REENTERPASSWORD)
    ibm_db.execute(prepare_stmt)

```

```

        return
render_template('Organization\Create_Account\org_create_account.
html')

    else:

        return
render_template('Organization\Create_Account\org_create_account.
html',error='Invalid Credentials')


return render_template('create_account.html',name='Home')


@app.route("/orgregister")
def orgregister():

    return render_template("org_create_account.html") #rendering
our register.html contained within /templates


@app.route("/user_dashboard")
def user_dashboard():

    return render_template("user_dashboard.html")


@app.route('/logout')
def logout():

    session.pop('email', None)

    return redirect(url_for('login'))

```

```
if __name__ == "__main__": #checking if __name__'s value is
'__main__'. __name__ is an python environment variable who's
value will always be '__main__' till this is the first instatnce
of app.py running
```

```
app.run(debug=True,port=8080) #running flask (Initalised on
line 4)
```

```
hash=bcrypt.hashpw(password.encode('utf-
8'),bcrypt.gensalt())
```

```
query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"
```

```
stmt = ibm_db.prepare(conn, query)
```

```
ibm_db.bind_param(stmt,1,email)
```

```
ibm_db.execute(stmt)
```

```
isUser = ibm_db.fetch_assoc(stmt)
```

```
if not isUser:
```

```
insert_sql = "INSERT INTO JOBSEEKER (FIRSTNAME,
LASTNAME,EMAIL_ADDRESSS,PWD,REPWD=?"
```

```
prep_stmt = ibm_db.prepare(conn, insert_sql)
```

```
ibm_db.bind_param(prepare_stmt, 1, firstname)
```

```
ibm_db.bind_param(prepare_stmt, 2, lastname)
```

```
ibm_db.bind_param(prepare_stmt, 3, email_address)
```

```
ibm_db.bind_param(prepare_stmt, 4, pwd)
```

```
ibm_db.bind_param(prepare_stmt, 5, repwd)
```

```

        ibm_db.execute(prepare_stmt)

        return render_template('Job Seeker/signup.html')

    else:

        return render_template('Job
Seeker/signup.html',error='Invalid Credentials')

return render_template('signup.html',name='Home')

```

JAVASCRIPT:

```

window.watsonAssistantChatOptions= {
    integrationID: "d73273d3-3f44-
4304-84ee-8fd243016d1d", // The ID of
this integration.
    region: "jp-tok", // The region
your integration is hosted in.
    serviceInstanceID: "81229104-ee6b-
46da-ac1c-67ed110663a", // 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.cli
entVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
        document.head.appendChild(t);
    });

```

Verify.py

```
from flask import request
from flask_restful import Resource
from utils.tokener import confirm_token
from utils.dbQuery import insertQuery

class Verify(Resource):
    data=request.json
    token=data["token"]
    email=confirm_token(token)
    if(not email):
        return {"status":"Couldn't verify"},400
    res=insertQuery("UPDATE user set verified=True where
email=?", (email,))
    if(not res):
        return {"status":"Couldn't Update"},400
    return {"status":"verified successfully"},200
```

SPRINT 3:

FRONT END

DASHBROAD.html

```
<!DOCTYPE html>

<html lang="en">

<head>

    <title>Bootstrap Example</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=device-width, initial-
scale=1">

    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst
rap.min.css">

    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>

    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra
p.min.js"></script>

<style>

    .rus{

        border-radius: 0%

    };

    .black{
```

```

        color:black;

    }
</style>

</head>

<body>

<!.....
=====-->

<nav class="navbar navbar-inverse rus">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">Skill and Job
Recommender</a>
        </div>

        <ul class="nav navbar-nav navbar-right">
            <li><a href="#"><span class="glyphicon glyphicon-log-
in"></span> Log out</a></li>
        </ul>
    </div>
</nav>

<div class="container">
    <ul class="nav nav-tabs">
        <li class="active"><a data-toggle="tab"
href="#home">Profile</a></li>

```



```

    <li><a data-toggle="tab" href="#menu1">Eligibility
Test</a></li>

    <li><a data-toggle="tab" href="#menu2">Find Job</a></li>

</ul>

<div class="tab-content">

    <div id="home" class="tab-pane fade in active">

        <h3>PROFILE</h3>

        <hr>

        <form>

            <div class="alert alert-danger">

                <div class="form-group">

                    <label for="name"><span class="black">Name:
</span></label>

                    <input type="text" class="form-control" id="name"
value="Mahadevan">

                </div>

                <div class="form-group">

                    <label for="email"><span class="black">Email Address
</span></label>

                    <input type="text" class="form-control" id="email"
value="Mahadevan@gmail.com">

                </div>

                <div class="form-group">

```

```
        <label for="email"><span class="black">Mobile No
</span></label>

        <input type="text" class="form-control" id="email"
value="9003824096">

    </div>

    <div class="form-group">

        <label for="quali"><span class="black">What is your
Nationality </span></label>

        <input type="text" class="form-control" id="quali"
value="India">

    </div>

    <div class="form-group">

        <label for="quali">What is your
Qualification</label>

        <input type="text" class="form-control" id="quali"
value="BE">

    </div>

    <div class="form-group">

        <label for="quali">What year did you complete your
bachelor Degree?</label>

        <input type="text" class="form-control" id="quali"
value="2023">

    </div>
```

```
<div class="form-group">

    <label for="experience">How many years of work
experience do you have ?</label>

    <input type="text" class="form-control"
id="experience" value="0">

</div>


<div class="form-group">

    <label for="place">Where have you graduated high
school degree ?</label>

    <input type="text" class="form-control" id="place"
value="Government high school Ariyalur">

</div>


<div class="form-group">

    <label for="SSLC">What is your SSLC Percentage
:</label>

    <input type="text" class="form-control" id="SSLC"
value="90">

</div>


<div class="form-group">

    <label for="HSC">What is your HSC Percentage
:</label>

    <input type="text" class="form-control" id="HSC"
value="90">

</div>
```

```

        <div class="form-group">

            <label for="degree">What is your Percentage in
bachelor Degree:</label>

                <input type="text" class="form-control" id="degree"
value="90.5">

            </div>


        <textarea class="form-control" id="aboutyou" rows="3"
placeholder="Write few words about yourself"></textarea> <br>
<br>


        <button type="submit" class="btn btn-primary btn-
success">Save changes</button> <br> <br>

    </div>

</form>

</div>

<div id="menu1" class="tab-pane fade">

    <h3>Eligibility Test </h3> <hr>


    <div class="alert alert-info">

        <label>What function is used to display output on the
screen?</label>

            <div class="radio">

                <label><input type="radio"
name="print">printf()</label>

            </div>

            <div class="radio">

```

```

        <label><input type="radio"
name="scanf">scanf()</label>

        </div>

        <div class="radio">

            <label><input type="radio"
name="switch">switch</label>

            </div>

    </div> <hr>

    <div class="alert alert-info">

        <label>What function is used to display output on the
screen?</label>

        <div class="radio">

            <label><input type="radio"
name="print">printf()</label>

            </div>

            <div class="radio">

                <label><input type="radio"
name="scanf">scanf()</label>

                </div>

                <div class="radio">

                    <label><input type="radio"
name="switch">switch</label>

                    </div>

            </div> <hr>

            <div class="alert alert-info">

```

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio"
name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"
name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"
name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio"
name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"
name="scanf">scanf()</label>

</div>

<div class="radio">

```
                <label><input type="radio"
name="switch">switch</label>
```

```
            </div>
```

```
    </div> <hr>
```

```
    <div class="alert alert-info">
```

```
        <label>What function is used to display output on the
screen?</label>
```

```
        <div class="radio">
```

```
            <label><input type="radio"
name="print">printf()</label>
```

```
        </div>
```

```
        <div class="radio">
```

```
            <label><input type="radio"
name="scanf">scanf()</label>
```

```
        </div>
```

```
        <div class="radio">
```

```
            <label><input type="radio"
name="switch">switch</label>
```

```
        </div>
```

```
    </div><hr>
```

```
    <div class="alert alert-info">
```

```
        <label>What function is used to display output on the
screen?</label>
```

```
        <div class="radio">
```

```
            <label><input type="radio"
name="print">printf()</label>
```

```

        </div>

        <div class="radio">

            <label><input type="radio"
name="scanf">scanf()</label>

        </div>

        <div class="radio">

            <label><input type="radio"
name="switch">switch</label>

        </div>

    </div>

    <button type="submit" class="btn btn-primary
btn-success">Submit the test</button> <br> <br>

</div>

<div id="menu2" class="tab-pane fade">

    <h3>Find the job</h3> <hr>

    <br>

    <div class="container">

        <form>

            <div class="input-group">

                <input type="text" class="form-control"
id="myInput" placeholder="Look for a job">

                <div class="input-group-btn">

```



```

        <p>Job Location : Chennai</p>

        <p> Salary : 45000</p>

        <button type="submit" class="btn btn-primary btn-
success">Apply </button> <br> <hr>

        </div>

    </div>

</div>

<script>

    $(document).ready(function() {

        $("#myInput").on("keyup", function() {

            var value = $(this).val().toLowerCase();

            $("#myDIV *").filter(function() {

                $(this).toggle($(this).text().toLowerCase().indexOf(value)
> -1)

            });

        });

    });

</script>

</body>

</html>

```

BACK END :

```
from flask import *#importing flask (Install it using python -m
pip install flask)

import os

import ibm_db

import bcrypt

#import send_from_directory

app = Flask(__name__) #initialising flask

app.secret_key = ''

PEOPLE_FOLDER = os.path.join('static', 'people_photo')

import ibm_db

conn = ibm_db.connect("DATABASE=bludb;
HOSTNAME=54a2f15b-5c0f-46df-8954-
7e38e612c2bd.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT
=32733;SECURITY=SSL;
SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID
=phb43134;PWD=mTTnIE45Raj9A0rr",'','')

@app.route("/") #defining the routes for the home() funtion
(Multiple routes can be used as seen here)

@app.route("/home")

def dashboard():

    return render_template("dashboard.html")

def createjobseeker_profile():

    if request.method == 'POST':

        name = request.form['name']

        email = request.form['email']

        mobileno=request.form['mobileno']
```

```

nationality=request.form['nationality']

qualification=request.form['qualification']

passout=request.form['passout']

experience=request.form['experience'];

degree=request.form['degree'];

per_sslc=request.form['per_sslc'];

per_hsc=request.form['per_hsc'];

per_degree=request.form['per_degree'];

tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD_DIR'],tum.filename))

hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"

stmt = ibm_db.prepare(conn, query)

ibm_db.bind_param(stmt,1,email)

ibm_db.execute(stmt)

isUser = ibm_db.fetch_assoc(stmt)

if not isUser:

    insert_sql = "INSERT INTO JOBSEEKER (name, email,
mobileno, nationality,
qualification, passout, experience,
degree,per_sslc,per_hsc,per_degree) VALUES
(?,?,?,?,?,?,?,?,?,?,?,?,?)"

    prep_stmt = ibm_db.prepare(conn, insert_sql)

    ibm_db.bind_param(prepare_stmt, 1, name)

    ibm_db.bind_param(prepare_stmt, 2, email)

```

```
    ibm_db.bind_param(prepare_stmt, 3, mobileno)
    ibm_db.bind_param(prepare_stmt, 4, nationality)
    ibm_db.bind_param(prepare_stmt, 5, qualification)
    ibm_db.bind_param(prepare_stmt, 6, passout)
    ibm_db.bind_param(prepare_stmt, 7, experience)
    ibm_db.bind_param(prepare_stmt, 8, degree)
    ibm_db.bind_param(prepare_stmt, 9, per_sslc)
    ibm_db.bind_param(prepare_stmt, 10, per_hsc)
    ibm_db.bind_param(prepare_stmt, 11, per_degree)
    ibm_db.execute(prepare_stmt)

    return render_template('dashboard.html')

return render_template('dashboard.html', name='Home')

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

SPRINT 4:

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>Bootstrap Example</title>

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-
scale=1">

  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst
rap.min.css">

  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m
in.js"></script>

  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra
p.min.js"></script>


  <style>

    .rus{

      border-radius: 0%

    };

    .black{

      color:black;

    }

  </style>
```

```

</head>

<body>

<!.....
=====-->

<nav class="navbar navbar-inverse rus">
    <div class="container-fluid">
        <div class="navbar-header">
            <a class="navbar-brand" href="#">Skill and Job
Recommender</a>
        </div>

        <ul class="nav navbar-nav navbar-right">
            <li><a href="#"><span class="glyphicon glyphicon-log-
in"></span> Log out</a></li>
        </ul>
    </div>
</nav>

<!.....
=====-->

<div class="container">
    <ul class="nav nav-tabs">
        <li class="active"><a data-toggle="tab" href="#home">Company
Profile</a></li>
        <li><a data-toggle="tab" href="#menu2">Find
candidate</a></li>

```

```

</ul>

<div class="tab-content">
  <div id="home" class="tab-pane fade in active">
    <h3>COMPANY PROFILE</h3>
    <hr>

    <form>
      <div class="alert alert-danger">
        <div class="form-group">
          <label for="name"><span class="black">Company Name:
</span></label>
          <input type="text" class="form-control" id="name"
value="Mahadevan">
        </div>
        <div class="form-group">
          <label for="email"><span class="black">Company Email
Address </span></label>
          <input type="text" class="form-control" id="email"
value="Mahadevan@gmail.com">
        </div>

        <div class="form-group">
          <label for="email"><span class="black">Mobile No
</span></label>
          <input type="text" class="form-control" id="email"
value="9003824096">

```

```
</div>
```

```
<div class="form-group">
```

```
  <label for="quali"><span  
class="black">Location</span></label>
```

```
    <input type="text" class="form-control" id="quali"  
value="India">
```

```
</div>
```

```
  <textarea class="form-control" id="aboutyou" rows="3"  
placeholder="Write few words about company"></textarea> <br>  
<br>
```

```
  <button type="submit" class="btn btn-primary btn-  
success">Save changes</button> <br> <br>
```

```
</div>
```

```
</form>
```

```
</div>
```

```
<div id="menu2" class="tab-pane fade">
```



```

<h3>Find the Condidate</h3> <hr>

<br>

<div class="container">

<!--.....

>

<form>

    <div class="input-group">

        <input type="text" class="form-control"
id="myInput" placeholder="Look for a candidate">

        <div class="input-group-btn">

            <button class="btn btn-default"
type="submit">

                <i class="glyphicon glyphicon-
search"></i>

            </button>

        </div>

    </div>

</form> <br> <br>

<!--.....>

<div style="width: 600px;">

    <!-- Left-aligned -->

    <div class="media">

        <div class="media-left">

```

```

        </div>

        <div class="media-body">

            <h4 class="media-heading">Name :
Mahadevan</h4>

            <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>

            <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>

            </div>

        </div>

    </div> <br>

<div style="width: 600px;">

    <!-- Left-aligned -->

    <div class="media">

        <div class="media-left">

            </div>

            <div class="media-body">

                <h4 class="media-heading">Name :
Manikandan</h4>

                <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>

                <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>

                </div>

```

```

        </div>

</div> <br>

<div style="width: 600px;">

    <!-- Left-aligned -->

    <div class="media">

        <div class="media-left">

            </div>

            <div class="media-body">

                <h4 class="media-heading">Name :
Kavi</h4>

                <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>

                <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>

            </div>

        </div>

    </div> <br>

<div style="width: 600px;">

    <!-- Left-aligned -->

    <div class="media">

        <div class="media-left">

```

```

        </div>

        <div class="media-body">

            <h4 class="media-heading">Name :
Ram</h4>

            <h5 class="media-heading"> Qualification : B.E
(Computer Science Engineering )</h4>

            <button type="submit" class="btn btn-primary btn-
success">Know More </button> <br> <hr>

        </div>

    </div>

</div>

<script>

    $(document).ready(function() {

        $("#myInput").on("keyup", function() {

            var value = $(this).val().toLowerCase();

            $("#myDIV *").filter(function() {

                $(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)

            });

        });

    });

</script>

</body>

</html>

```

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-53465-1661409305.git>