

Skill / Job Recommender Application

IBM-Project-42576-1660668622

PROJECT REPORT

SUBMITTED BY

TEAM ID : PNT2022TMID33378

S.MUTHU GANESH	(922519104098)
R.NARESH	(922519104102)
S.PRASATH KUMAR	(922519104118)
S.PRAVEEN KUMAR	(922519104119)

*In partial fulfilment for the award of the degree
Of*

BACHELOR OF ENGINEERING

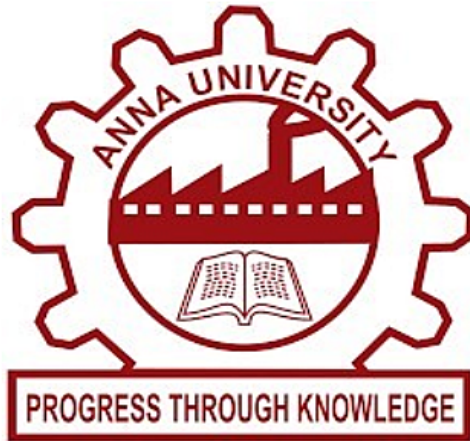
In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



V.S.B.ENGINEERING COLLEGE ,KARUR - 639111

(Approved by AICTE & Affiliated by Anna University, Chennai)



BONAFIDE CERTIFICATE

Certified that this mini project report titled "Skill and Job Recommendation " is the bonafide record work by

S.Muthuganesh(922519104098),R.Naresh(922519104102), S.Prasath kumar(922519104118) and S.Praveen kumar(922519104119)

for **IBM-NALAIYATHIRAN** in VII semester of B.E., degree course in Computer Science and Engineering branch during the academic year of 2022-2023.

**Staff-In
Charge
Manikandan.TP**

**Evalutor
Gunasekaran**

**Head of the Department
Mr. Anbumani P**

ACKNOWLEDGEMENT

First and foremost, we express my thanks to our parents for providing us a very nice environment for doing this mini project. We wish to express our sincere thanks to our founder and **Chairman Shri.V.S.BALSAMY** for his endeavor in educating us in this premier institution.

We wish to express our appreciation and gratefulness to our principal, **Dr.V.NIRMAL KANNAN** and vice principal **Mr.T.S.KIRUBASANKAR** for their encouragement and sincere guidance.

We are grateful to our head of the department **Mr.P.ANBUMANI** and our Nalaiyathiran project coordinator **Mr.P.GUNASEKARAN** Department of Computer Science and Engineering for their valuable support.

We express our indebtedness to the supervisor of our Nalaiyathiran project, **Mr.P.GUNASEKARAN** Assistant Professor, Department of Computer Science and Engineering, for guidance throughout the course of our project.

Our sincere thanks to all the teaching staff of V.S.B Engineering College and our friends for their help in the successful completion of this IBM Nalaiyathiran project work. Finally, we bow before God, the almighty who always had a better plan for us. We give our praise and glory to Almighty God for successful completion of this IBM Nalaiyathiran.

INDEX

1. INTRODUCTION

- a. Project Overview
- b. Purpose

2. LITERATURE SURVEY

- a. Existing problem
- b. References
- c. Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

- a. Empathy Map Canvas
- b. Ideation & Brainstorming
- c. Proposed Solution
- d. Problem Solution fit

4. REQUIREMENT ANALYSIS

- a. Functional requirement
- b. Non-Functional requirements

5. PROJECT DESIGN

- a. Data Flow Diagrams
- b. Solution & Technical Architecture

6. PROJECT PLANNING & SCHEDULING

- a. Sprint Planning & Estimation
- b. Sprint Delivery Schedule
- c. Reports from JIRA

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

with code)

- a. Feature 1
- b. Feature 2

8. TESTING

- a. Test Cases
- b. User Acceptance Testing

9. RESULTS

- a. Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11. CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code
GitHub & Project Demo Link

1. INTRODUCTION

1.1 Project 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.

1.2 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 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 Indiana University Purdue University Indianapolis vsdave@iupui.edu Baichuan Zhang Facebook Inc baichuan24@fb.com Mohammad Al Hasan Indiana University Purdue University Indianapolis alhasan@iupui.edu Khalifeh AlJadda and Mohammed Korayem CareerBuilder, Inc, Norcross, GA khalifeh.aljadda,mohammed.korayem@careerbuilder.com

2. Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation Akshay Gugnani,¹ Hemant Misra^{2*} ¹IBM Research - AI, ²Applied Research, Swiggy, India aksgug22@in.ibm.com, hemant.misra@swiggy.in

3. Job Recommendation System Using Profile Matching And Web-Crawling Deepali V Musale ¹ , Mamta K Nagpure² , Kaumudini S Patil³ , Rukhsar F Sayyed⁴ Students, Computer Science & Engineering, K K Wagh College of Engineering, Nashik,

India1 2 3 4

4. Job Seekers' Acceptance of Job Recommender Systems: Results of an Empirical Study
Sven Laumer University of Bamberg sven.laumer@uni-bamberg.de
Fabian Gubler University of Bamberg fabian.t.gubler@gmail.com
Christian Maier University of Bamberg christian.maier@uni-bamberg.de
Tim Weitzel University of Bamberg tim.weitzel@unibamberg.de
[1.https://www.researchgate.net/publication/272802616_A_survey_of_job_recommender_systems](https://www.researchgate.net/publication/272802616_A_survey_of_job_recommender_systems)

2.3 Problem Statement Definition :

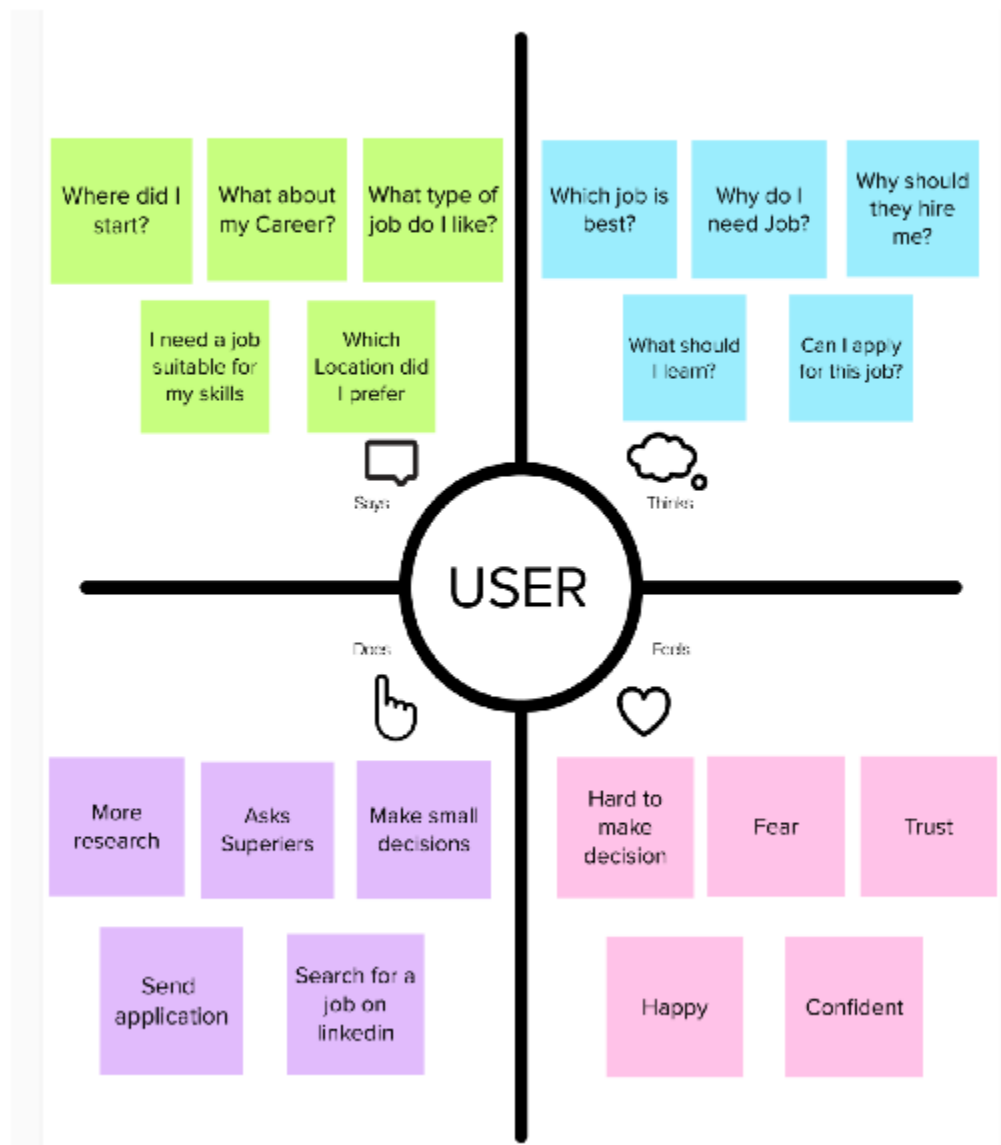
Mr. Muthuganesh is an engineering graduate who completed his graduation in xyz college who needs a job to match his skills. He has applied for several company selection process. But the problem is he does not get a job that match his skills. In most cases, he will get eliminated in the technical rounds as the technical rounds' questions are asked based on the job roles. If he has a skill set regarding web development, he cannot clear the technical interview on other topics such as CRM, graphics designing, dot net etc. So, he needs something which should save his time by not going in person to apply for a job. As, going in person will include the traveling time and waiting time. So, an application based on the internet will contain all the companies' information about the selection process and the role they are recruiting. So that he can apply from everywhere and what roles they are applying for.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:

An **empathy map** is a widely-used visualization tool within the field of UX and HCII practice. In relation to empathetic design, the primary purpose of an empathy map is to bridge the understanding of the end user. Within context of its application, this tool is used to build a shared understanding of the user's needs and provide context to a user-centered solution

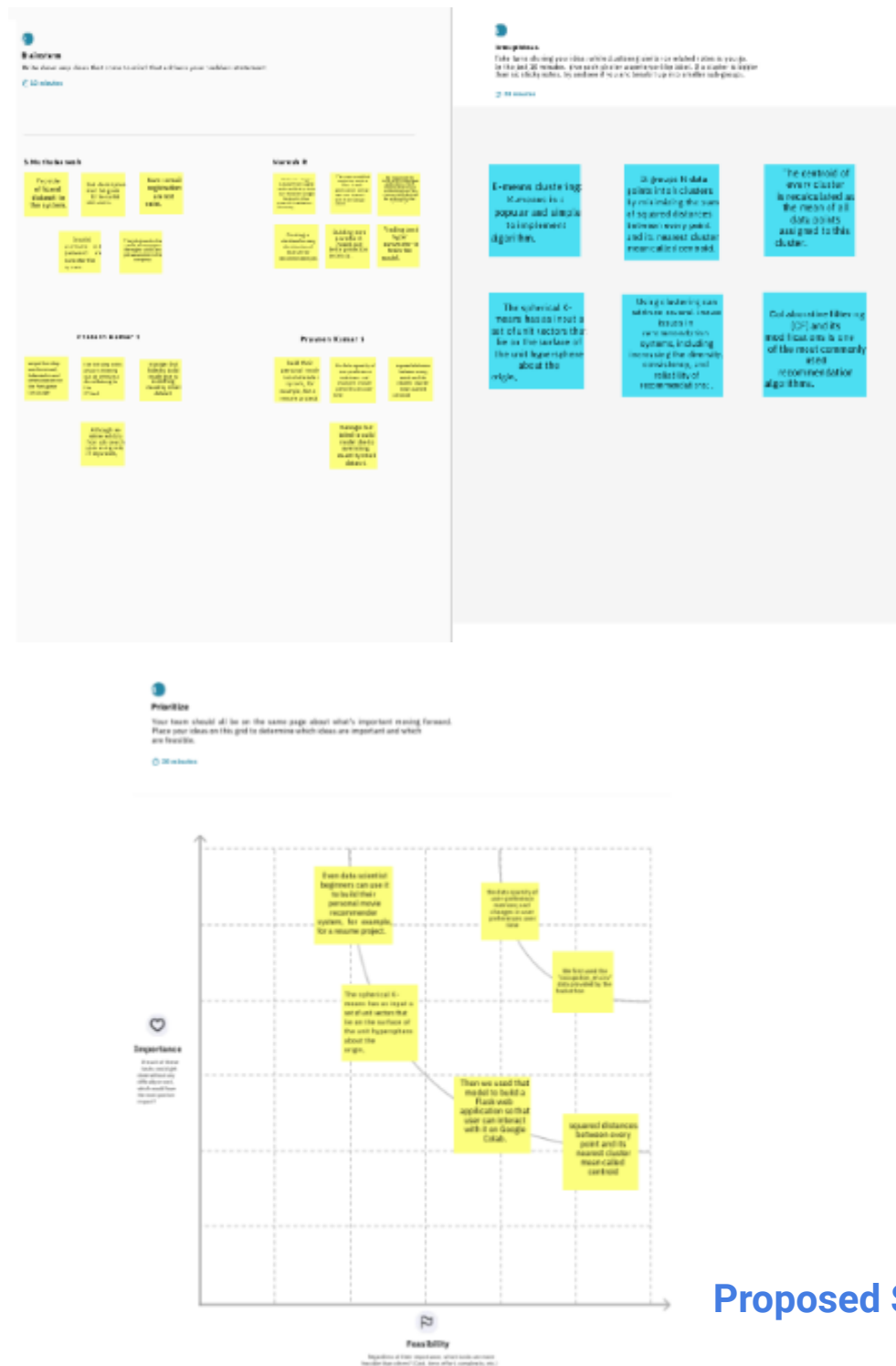
Empathy Map :



3.2

3.3

Proposed Solution



Team ID PNT2022TMID22892

PROJECT NAME SKILL & JOB RECOMMENDER

The project team shall fill the following information in proposed solution template.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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.
2.	Idea / Solution description	Start with an audience-first focus. Instead of thinking about what your organization produces (job availability) and how to fit that into an app, start from scratch and consider your audience. Who are they, and what problems do they

		face? Use your imagination, do some role playing with colleagues, and actually talk to some real people.
3.	Novelty / Uniqueness	Building mobile tools with data isn't as easy as importing an XML feed of your latest headlines. But if you're going to spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problems and makes advertisers take notice.
4.	Social Impact / Customer Satisfaction	These apps are all about solving problems and enabling the user to take an action — go to this bar, shop at that boutique, rent this apartment. These are the apps that build loyal audiences because they help people get things done instead of just presenting another thing they feel obligated to do (keep up with news stories). That is valuable not only for users, but for advertisers as well.

5.	Business Model (Revenue Model)	Building mobile tools with data isn't as easy as importing an XML feed of your latest available jobs. But if you're going to spend thousands of dollars developing a mobile app anyway, you might as well spend a little more to build a real application that helps solve problems and makes advertisers take notice.
----	--------------------------------	--

3.4 Problem Solution Fit :

Problem-Solution fit canvas 2.0

Define CS, fit into CC	1. CUSTOMER SEGMENT(S) CS Graduated Students Ex: Engineers, Employees	6. CUSTOMER CONSTRAINTS CC not having deep knowledge in technology, network connection, not having enough skills,	5. AVAILABLE SOLUTIONS AS Obtaining job availability information from the newspaper and searching for trending skills and available jobs.	Explore AS, differentiate
	2. JOBS-TO-BE-DONE / PROBLEMS J&P There could be more than one job suggestions and skill suggestions for graduated students	9. PROBLEM ROOT CAUSE RC It has become difficult to get a job by checking the availability of jobs and not having the necessary skills that companies are looking for.	7. BEHAVIOUR BE User needs to install this application from a verified server. Needs internet facility Throughout this application to learn skills and search jobs.	
Identify strong TR & EM	3. TRIGGERS TR seeing their neighbors and friends are getting good quality of job suggestions from the job recommender.	10. YOUR SOLUTION SL 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.	8. CHANNELS OF BEHAVIOUR CH ONLINE Client can do everything in this application utilizing on the web OFFLINE Client can only able to learn skills by downloading contents, searching job is difficult in offline	Extract online & offline CH of BE
	4. EMOTIONS: BEFORE / AFTER EM Anywhere at any time- This Application can be useful and can be used whenever we want.			

4. REQUIREMENT ANALYSIS

4.1 Function Requirement :

Software Required :

- Python
- Flask
- Docker

4.2 Non-Function Requirement:

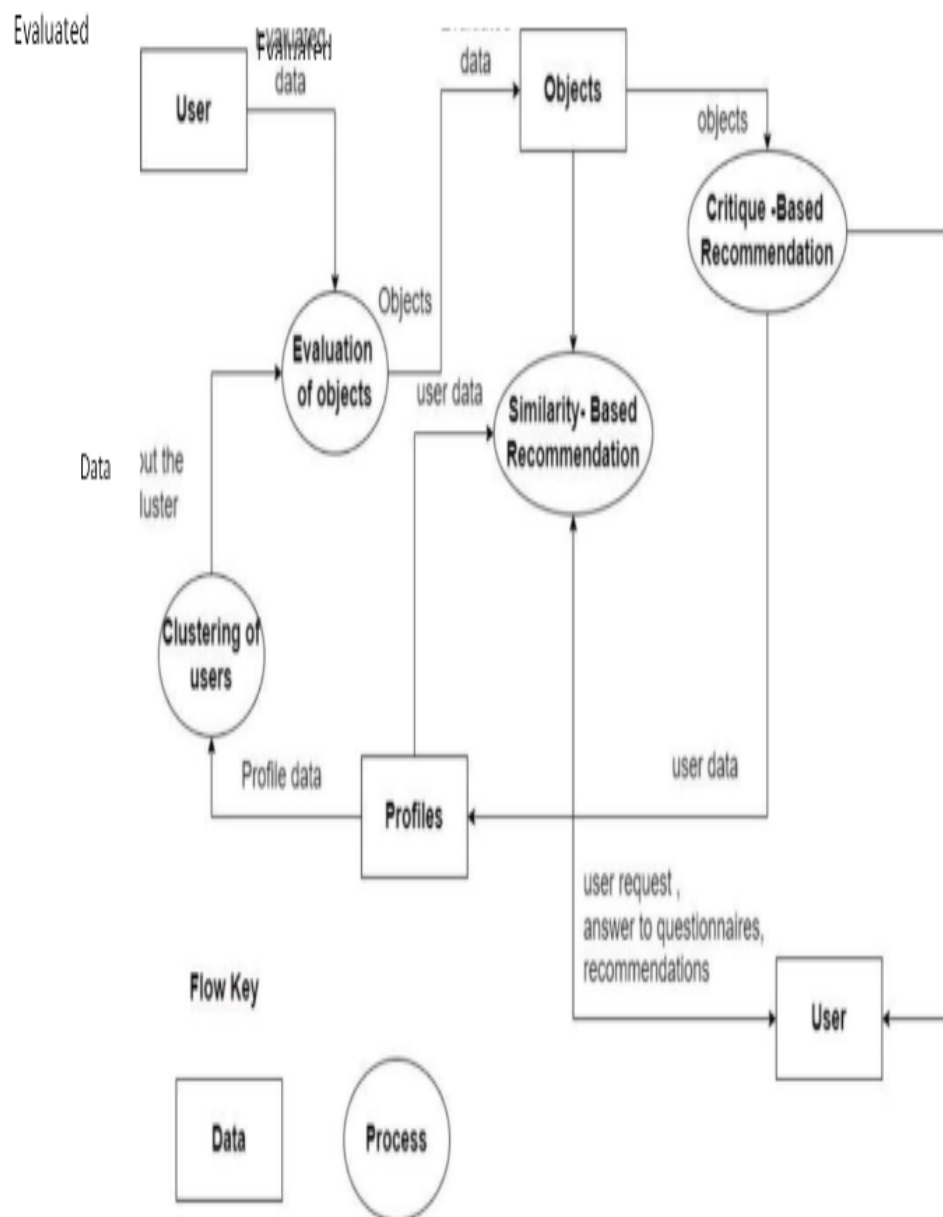
System Required :

8GB RAM, Intel CoreWindows/Linux/MAC,Laptop or Desktop

5. PROJECT DESIGN

5.1 Data Flow Diagrams

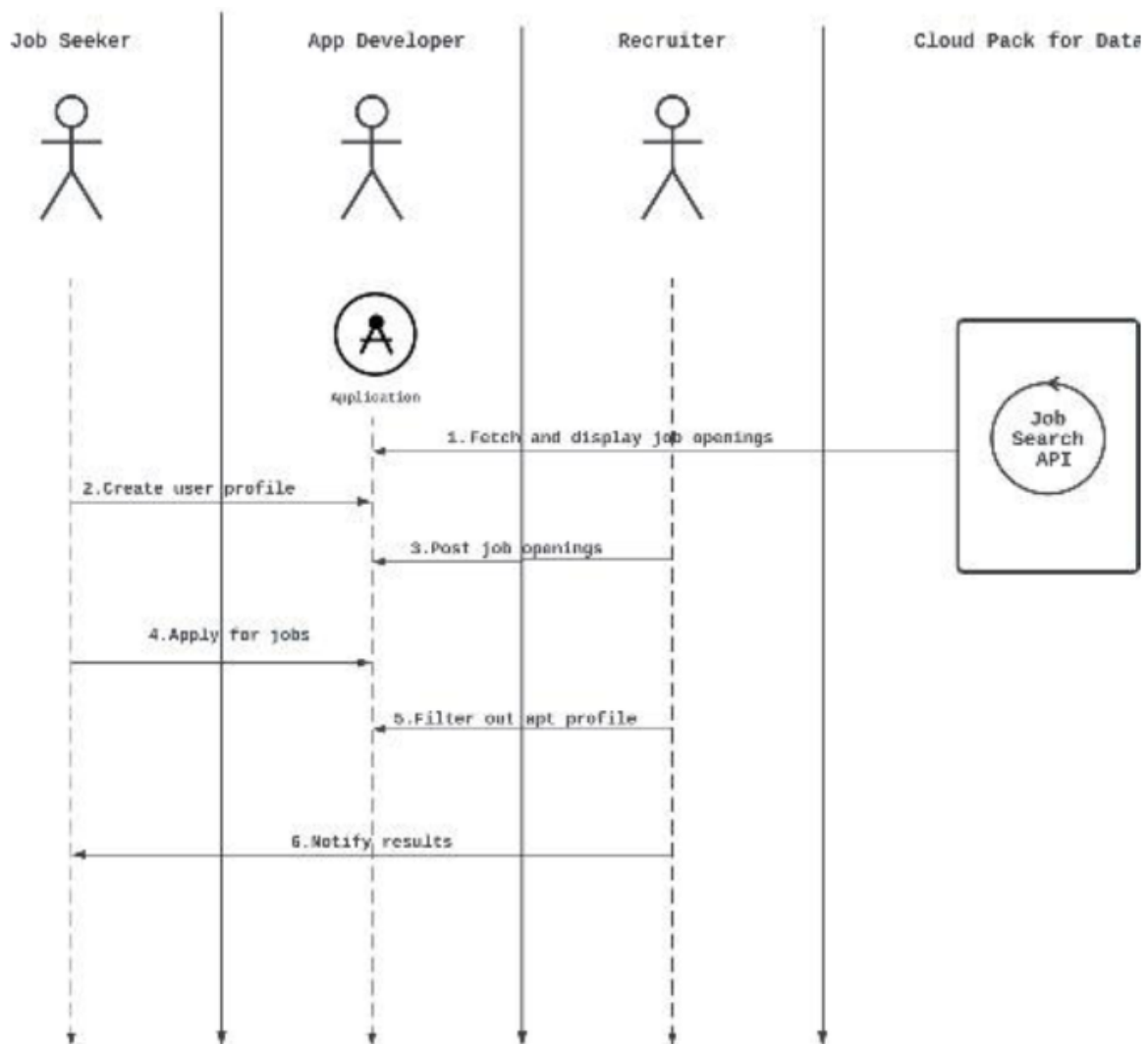
Data Flow Diagram:



about the user's
cluster

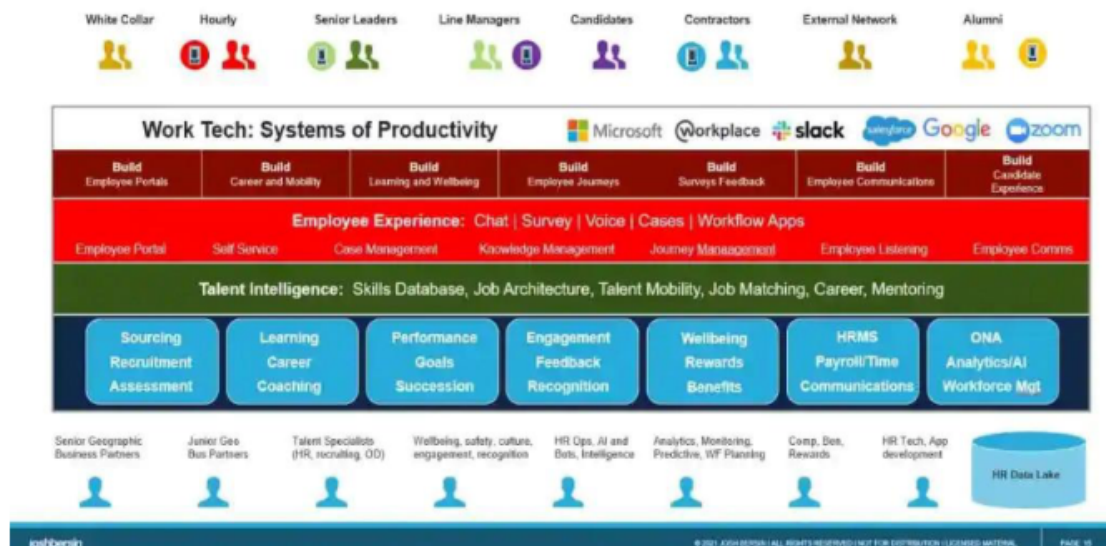
5.2 Solution & Technical Architecture:

SOLUTION ARCHITECTURE



TECHNICAL ARCHITECTURE :

HR Technology Architecture: 2021 and Beyond



Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant

5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud ant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used

4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

6. PROJECT PLANNING & SCHEDULING

Project scheduling involves creating a detailed project agenda that organizes all project to-dos and their deadlines. Project schedules work in tandem with a work breakdown structure (WBS), which breaks down projects into smaller achievable milestones. According to [PMI](#), WBS organizes and defines a project's scope by decomposing it into work packages.

Managing a project without a schedule is like building your dream house with just a picture, basic tools and a nagging question: where do we even start? Immaculate project scheduling answers all your basic questions, resolves uncertainties and provides a clear path to the finish line.

Define all your project tasks, start and end dates, workflow cycle and resources needed to cover the project scope. It also monitors team workflows by offering you visibility into every individual's progress. The process eventually helps you optimize workload management.

Notably, it is never a static process. You will have to add or

remove tasks, redefine milestones if they take longer than expected, and update schedules continuously. Comprehensive schedules keep you at the helm of all deliverables in the course of a project's life cycle.

6.1 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	MUTHU GANESH S PRAVEEN KUMAR S
Sprint-1		USN-2	As a user, I will receive confirmation email once I have registered for the	I can receive confirmation email & click confirm	High	NARESH R

			application			
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	PRASATH KUMAR S
Sprint-1		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	MUTHU GANESH S
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	NARESH R
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	PRAVEEN KUMAR S

Sprint-3	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	MUTHU GANESH S
Sprint-4	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	PRASATH KUMAR S
Sprint-4	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the	Ease of use.	Medium	PRAVEEN KUMAR S

			directory.			
--	--	--	------------	--	--	--

6.2 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	30 Oct 2022	04 Nov 2022	20	4 Nov 2022
Sprint-3	20	6 Days	05 Nov 2022	10 Nov 2022	20	10 Nov 2022
Sprint-4	20	6 Days	11 Nov 2022	16 Nov 2022	20	16 Nov 2022

6.3 Report from JIRA :

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

Common Pitfalls :

A Velocity measurement can only be developed after at least three sprints and is a measurement made after the fact in order to plan ahead, this therefore takes time which can sometimes anger stakeholder.

Velocity often differs between Sprints so it not a consistent measurable throughout projects and it is not project transferable and does not have a meaning comparison between different project teams.

Advantages of velocity :

Velocity aids delivery teams become more efficient and in terms of sprint planning and overall delivery and can decrease sprints by 10%.

Velocity is a useful tool to analyse key changes within a project, and whether change have had a positive or negative effect on the project velocity that can than be quickly amended. Velocity is a key part in team of release planing which helps keep key stakholder informed in term of

expected project progression and delivery

7. CODING & SOLUTIONING

7.1 Feature 1 JOB SEEKER:

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/boo
tstrap.min.css">
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquer
y.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/boots
trap.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">
<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</h4>
<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>
<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</h4>
<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</h4>
<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</h4>
<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)
});
});
});
});
</script>
</body>
```

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;S
SLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL
=TCPIP;UID=phb43134;PWD=mTTnIE45 Raj9A0rr",'','')
@app.route("/") #defining the routes for the home()
function (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']
```

```
mobilenumber=request.form['mobilenumber']
nationality=request.form['nationality']
qualification=request.form['qualification']
password=request.form['password']
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,
mobilenumber, nationality, qualification, password,
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'))
```

7.2 Feature 2 :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width,
initialscale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/boo
tstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquer
y.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/boots
trap.min.js"></script>
  <style>
    .rus{ border-radius:
      0%
    };
  </style>
```



```
.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-
login"></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
btnsuccess">Save changes</button> <br> <br>
```

```
    </div>
```

```
</form>
```

```
</div>
```

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

```
    <h3>Find the Candidate</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
btnsuccess">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

```

```

btnsuccess">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
btnsuccess">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
btnsuccess">Know More </button> <br> <hr>
            </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>
```

CHATBOT :

Chatbot has been implemented to provide assistance .

7.3 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? "))

```

8. TESTING

8.1 Test Cases

				Date	05-Nov-22									
				Team ID	PNT2022TMID33378									
				Project Name	Project - Skill/Job Recommender Application									
				Maximum Marks	4 marks									
Test case ID	Feature Type	Component	Test Scenario	Pre-Requirement	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By	
LoginPage_TC_O1	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/1OftuadB8X7eoVjyCMq4t3Nm936f5	Users should be navigated to registration page	Working as expected	Pass	Excellent	N		MUTHUGANESH S	
LoginPage_TC_O2	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/1Dkpi123P1nT2Ck5wZyFA8BELGvY9Rak	Users should be notified of login process	Not working	Fail	Trying To Recover	N	BUG12	NARESH R	
LoginPage_TC_O3	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		PRAVEENKUMAR S	
HomePage_TC_O1	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	BUG14	MUTHUGANESH S	
HomePage_TC_O2	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		PRAASATHKUMAR S	
HomePage_TC_O3	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		NARESH R	

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:

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

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

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

9. RESULTS

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

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 addition, Chatbot has been implemented with the help of IBM Watson. The chatbot helps jobseeker and organization when they experience the difficulties.

12. FUTURE SCOP

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

Sprint 1:

```
from flask import Flask,
render_template,request,redirect,url_for,session,flash
import ibm_db

app=Flask(__name__)
app.secret_key='a'
try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=125f9f61-
9715-46f9-9399-
c8177b21803b.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=
30426;SECURITY=SSL;SSLServerCertificat=DigiCertGlobalRootCA.crt;UI
D=wfq61013;PWD=mlp4pZJpldWruPNU","")
except:
    print("Unable to connect: ",ibm_db.conn_error())

@app.route("/")
```

```
def dash():
    return render_template('welcome.html',msg=" ")

@app.route("/register",methods=['GET','POST'])
def register():
    error = None
    if request.method=='POST':
        username=request.form['username']
        email=request.form['email']
        phone_number=request.form['phonenumber']
        password=request.form['password']
        pin=request.form['pin']
        sql="SELECT * FROM user WHERE phone_number=?"
        prep_stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(prepare_stmt,1,phone_number)
        ibm_db.execute(prepare_stmt)
        account=ibm_db.fetch_assoc(prepare_stmt)
        print(account)
        if account:
            error="Account already exists! Log in to continue !"
        else:
            insert_sql="INSERT INTO user values(?,?,?,?,?)"
            prep_stmt=ibm_db.prepare(conn,insert_sql)
            ibm_db.bind_param(prepare_stmt,1,email)
            ibm_db.bind_param(prepare_stmt,2,username)
            ibm_db.bind_param(prepare_stmt,3,phone_number)
            ibm_db.bind_param(prepare_stmt,4,password)
            ibm_db.bind_param(prepare_stmt,5,pin)
            ibm_db.execute(prepare_stmt)
```

```
        flash(" Registration successfull. Log in to continue !")
    else:
        pass
    return render_template('register.html',error=error)
```

```
@app.route('/login',methods=['GET','POST'])
def login():
    error = None
    if request.method=='POST':
        username=request.form['username']
        password=request.form['password']
        sql="SELECT * FROM user WHERE username=? AND password=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session["username"]=account["USERNAME"]
            flash("Logged in successfully!")
            return redirect(url_for("home"))
        else:
            error="Incorrect username / password"
            return render_template('login.html',error=error)
    return render_template('login.html',error=error)
```



```
@app.route('/forget',methods=['GET','POST'])
def forget():
    error = None
    if request.method=='POST':
        username=request.form['username']
        pin=request.form['pin']
        sql="SELECT * FROM user WHERE username=? AND pin=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,pin)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session['username']=account['USERNAME']
            flash("Logged in successfully!")
            return redirect(url_for("home"))
        else:
            error="Incorrect username / pin"
            return render_template('login.html',error=error)
    return render_template('forget.html',error=error)

@app.route('/welcome')
def welcome_page():
    return render_template("welcome.html",msg=" ")

@app.route('/home')
def home():
    return render_template("home.html",msg=" ")
```

```
if __name__=='__main__':  
    app.run(debug=True)
```

Sprint 2:

```
<html>  
  <head>  
    <title>Login page</title>  
    <link rel="stylesheet" href="{{url_for('static',  
filename='style.css')}}">  
  </head>  
  <body>  
    <header>  
      <nav class="navbar">  
        <ul>  
          <li><a href="{{url_for('welcome_page')}}">Back to  
page</a></li>  
        </ul>  
      </nav>  
    </header>  
    <section id="sss1">  
      <center><h1 class="two" ><span style="color:rgb(109, 60,  
225);background-color: whitesmoke;margin-left: 30px;padding-left:  
20px;padding-right: 20px;">SKYS</span>.com</h1></center>  
    </section>  
    <br><br>  
    <div class="container"> <br /><br />  
    <form action="/login" method="POST">  
      <h1>Login</h1> <br /><br />  
      <label
```

```

class="form_label"for="email"><b>Username</b></label><br><br>
    <input class="form_input"type="text" name= "username"
/><br><br>
    <label
class="form_label"for="psw"><b>Password</b></label><br><br>
    <input class="form_input"type="password"
name="password"/>
    </br></br></br>
    <center><input type="submit"
class="submitbtn"value="submit" /></center>
    </form>
</div>
    <center><p>Forgot your password? <a
href="{{url_for('forget')}}">Try Another Way</a></p></center>
    <center><p>Don't have a account <a
href="{{url_for('register')}}">Create new account</a></p></center>
</body>
</html>

```

Sprint 3:

```

<html>
    <head>
        <title>Login page</title>
        <link rel="stylesheet" href="{{url_for('static',
filename='style.css')}}">
    </head>
    <body>
        <header>

```

```

<nav class="navbar">
  <ul>
    <li><a href="{{url_for('welcome_page')}}">Back to
page</a></li>
  </ul>
</nav>
</header>
<section id="sss1">
  <center><h1 class="two" ><span style="color:rgb(109, 60,
225);background-color: whitesmoke;margin-left: 30px;padding-left:
20px;padding-right: 20px;">SKYS</span>.com</h1></center>
</section>
<br><br>
<div class="container"> <br /><br />
<form action="/login" method="POST">
  <h1>Login</h1> <br /><br />
  <label
class="form_label"for="email"><b>Username</b></label><br><br>
    <input class="form_input"type="text" name= "username"
/><br><br>
    <label
class="form_label"for="psw"><b>Password</b></label><br><br>
    <input class="form_input"type="password"
name="password"/>
    </br></br></br>
    <center><input type="submit"
class="submitbtn"value="submit" /></center>
  </form>
</div>

```

```
<center><p>Forgot your password? <a
href="{{url_for('forget')}}">Try Another Way</a></p></center>
<center><p>Don't have a account <a
href="{{url_for('register')}}">Create new account</a></p></center>
</body>
</html>
```

Sprint 4:

Templates:

about.html :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SKYS.com</title>
  <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Home</a></li>
        <li><a href="{{url_for('register')}}">Register</a></li>
        <li><a href="{{url_for('login')}}">Login</a></li>
```

```
        </ul>
    </nav>
</header>
<section id="sss1">
    <center><h1 class="two" ><span style="color:rgb(20, 20,
70);background-color: whitesmoke;margin-left: 30px;padding-left:
20px;padding-right: 20px;">SKYS.com</span></h1></center>
</section>
</body>
</html>
```

Contact.htm

Create UI interact with web Application :

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <!--stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/font-awesome.min.css">
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.c
ss">
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></scri
pt>
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">
```

```
</script>
<style>
  span::before {
    content: "Android Developers";
    animation: animate infinite 3s;
    padding-left: 10px;
  }

  @keyframes animate {

    0% {
      content: "Android Developers";
      color:rgb(128, 0, 53);
    }

    50% {
      content: "NodeJs Developers";
      color:rgb(128, 0, 53);
    }

    75% {
      content: "Python Developers";
      color:rgb(128, 0, 53);
    }
    80% {
      content: "Java Developers";
      color:rgb(128, 0, 53);
    }
  }
}
```

```
* {  
  box-sizing: border-box;  
}
```

```
body {  
  font-family: bodoni mt;  
  padding: 10px;  
  background-image: url('Images/bg.webp');  
  background-color: black;  
  color:white;  
}
```

```
/* Header/Blog Title */
```

```
.header {  
  padding: 30px;  
  text-align: center;  
  background-image: url('Images/bg1.webp');  
  background-color: black;  
}
```

```
.header h1 {  
  font-size: 50px;  
}
```

```
/* Style the top navigation bar */
```

```
.topnav1 {  
  overflow: hidden;
```



```
background-color: rgb(0, 0, 0);  
}
```

```
/* Style the topnav1 links */
```

```
.topnav1 a {  
  float: left;  
  display: block;  
  color: #f2f2f2;  
  text-align: center;  
  padding: 14px 16px;  
  text-decoration: none;  
}
```

```
/* Change color on hover */
```

```
.topnav1 a:hover {  
  background-color: rgb(255, 255, 255);  
  color: rgb(177, 34, 75);  
}
```

```
/* Create two unequal columns that floats next to each other */
```

```
/* Left column */
```

```
.leftcol {  
  float: left;  
  width: 75%;  
}
```

```
/* Right column */
```

```
.rightcol {  
  float: left;
```

```
width: 25%;  
background-color: #f1f1f1;  
padding-left: 20px;  
}
```

```
/* Fake image */  
.fakeimage {  
  background-color: #aaa;  
  width: 100%;  
  padding: 20px;  
}
```

```
/* Add a card effect for articles */  
.card {  
  background-color: rgb(0, 0, 0);  
  padding: 20px;  
  margin-top: 20px;  
  color: white;  
}  
.card1 {  
  background-color: rgb(0, 0, 0);  
  padding: 20px;  
  margin-top: 20px;  
  align-self: center;  
  color:white  
}  
/* Clear floats after the columns */  
.row:after {  
  content: "";
```

```
display: table;
clear: both;
}
```

```
/* Footer */
```

```
.footer {
padding: 20px;
text-align: center;
background: #ddd;
margin-top: 20px;
}
```

```
* {
box-sizing: border-box;
}
```

```
input[type=text], select, textarea {
width: 100%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 4px;
resize: vertical;
}
```

```
input[type=password], select, textarea {
width: 100%;
padding: 12px;
border: 1px solid #ccc;
border-radius: 4px;
resize: vertical;
}
```

```
}
```

```
label {  
  padding: 12px 12px 12px 0;  
  display: inline-block;  
}
```

```
input[type=submit] {  
  background-color: #04AA6D;  
  color: white;  
  padding: 12px 20px;  
  border: none;  
  border-radius: 4px;  
  cursor: pointer;  
  float: right;  
}
```

```
input[type=submit]:hover {  
  background-color: #45a049;  
}
```

```
.registerbtn {  
  background-color: #000000d2;  
  color: white;  
  padding: 16px 20px;  
  margin: 8px 0;  
  border: none;  
  cursor: pointer;  
  width: 100%;  
  opacity: 0
```

```
}  
.container {  
  border-radius: 5px;  
  background-image: url('Images/bg1.webp');  
  background-color: black;  
  padding: 20px;  
}
```

```
.col-25 {  
  float: left;  
  width: 25%;  
  margin-top: 6px;  
}
```

```
.col-75 {  
  float: left;  
  width: 75%;  
  margin-top: 6px;  
}
```

```
/* Clear floats after the columns */  
.row:after {  
  content: "";  
  display: table;  
  clear: both;  
}
```

/* Responsive layout - when the screen is less than 800px wide, make

the two columns stack on top of each other instead of next to each other */

```
@media screen and (max-width: 800px) {  
  .leftcol, .rightcol {  
    width: 100%;  
    padding: 0;  
  }  
}
```

/* Responsive layout - when the screen is less than 400px wide, make the navigation links stack on top of each other instead of next to each other */

```
@media screen and (max-width: 400px) {  
  .topnav1 a {  
    float: none;  
    width: 100%;  
  }  
}
```

</style>

</head>

<body>

<div class="header">

<h1>Skill And Job Recommender</h1>

<h1>Inspiring Jobs for the future !</h1>

</div>

<div class="topnav1">

<i class="fa fa-home" style="font-size:24px"></i><pre>Home</pre>

<i class="fa fa-sign-in" style="font-

```

size:24px"></i><pre>Login</pre></a>
    <a href="register.html"><i class="fa fa-registered" style="font-
size:24px"></i><pre>Register</pre></a>
    <a href="home.html"><i class="fa fa-eye" style="font-
size:24px"></i><pre>About Us</pre></a>
    <a href="job.html"><i class="fa fa-graduation-cap" style="font-
size:24px"></i><pre>Job Chooser</pre></a>
    <a href="#" style="float:right"><i class="fa fa-sign-out" style="font-
size:24px"></i><pre>Sign Out</pre></a>
</div>
<div class="container">
<form action="/action_page.php">
    <div class="row">
        <div class="col-25">
            <label for="fname">Email </label>
        </div>
        <div class="col-75">
            <input type="text" id="fname" name="firstname" placeholder="Email
Id">
        </div>
    </div>
    <div class="row">
        <div class="col-25">
            <label for="lname">Password</label>
        </div>
        <div class="col-75">
            <input type="password" id="lname" name="lastname"
placeholder="Your last name..">
        </div>
    </div>

```

```
</div>
<div class="row">
  <div class="col-25">
    <label for="lname">Choose Your skill</label></div>
    <div class="col-75">
      <input type="checkbox" id="java" name="java">
      <label for="java">Java</label><br>
      <input type="checkbox" id="C++" name="C++">
      <label for="C++">C++</label><br>
      <input type="checkbox" id="Python" name="Python" >
      <label for="Python">Python</label><br>
      <input type="checkbox" id="DevOps" name="DevOps">
      <label for="Python">DevOps</label><br>
      <input type="checkbox" id="NodeJs" name="NodeJs">
      <label for="Python">Node Js</label><br>
      <input type="checkbox" id="ML" name="ML">
      <label for="Python">Machine Learning</label><br>
      <input type="checkbox" id="Civil" name="Civil">
      <label for="Python">Structural analysis</label><br>
      <input type="checkbox" id="Robotics" name="Robotics">
      <label for="Python">Robotics</label><br>
      <input type="checkbox" id="Full_Stack" name="Full_Stack">
      <label for="Python">Full Stack</label><br>
      <input type="checkbox" id="PHP" name="PHP">
      <label for="Python">PHP</label><br>
      <input type="checkbox" id="UiUx" name="UiUx">
      <label for="Python">UI/UX</label><br>
      <input type="checkbox" id="Automation" name="Automation">
      <label for="Python">Automation</label><br>
```



```
<br></div>
</div>
<div class="row">
  <div class="col-25">
    <label for="subject">Mention Your Skills If Any</label>
  </div>
  <div class="col-75">
    <textarea id="subject" name="subject" placeholder="Skills Here....."
style="height:200px"></textarea>
  </div>
</div>
<div class="registerbtn">
<button>Submit</button>
</div>
</form>
</div>
</body>
</html>
```

Contact.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>SKYS.com</title>
  <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
```

```
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Home</a></li>
        <li><a href="{{url_for('register')}}">Register</a></li>
        <li><a href="{{url_for('login')}}">Login</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <center><h1 class="two" ><span style="color:rgb(20, 20,
70);background-color: whitesmoke;margin-left: 30px;padding-left:
20px;padding-right: 20px;">SKYS.com</span></h1></center>
  </section>
  <br><br>
  <center><h1 class="h1lt">Contact us</h1></center>

  <form>
  First name:<br>
  <input type="text" name="firstname" value="">
  <br>
  Last name:<br>
  <input type="text" name="lastname" value="">
  <br>
  Email:<br>
  <input type="text" name="email" value="">
  <br>
```

```
<br>
<textarea name="message" rows="10" cols="30">
</textarea>
<br><br>
<input type="submit" value="Submit">
</form>

</body>
</html>
```

forget.html

```
<html>
  <head>
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
      <nav class="navbar">
        <ul>
          <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
        </ul>
      </nav>
    </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(109, 30,
192);background-color: whitesmoke;margin-left: 30px;padding-left:
20px;padding-right: 20px;">SKYS</span>.com</h1></center>
```

```

</section>
<br><br>
{% if error %}
    <p><strong style="color:red">Error</strong>: {{error}}</p>
{% endif %}
{% with messages = get_flashed_messages() %}
    {% if messages %}
        {% for message in messages %}
            <p style="color:green">{{ message }}</p>
        {% endfor %}
    {% endif %}
{% endwith %}
<div class="container"> <br /><br />
<form action="/forget" method="POST">
    <h1>Try to login with your 4 digit pin</h1> <br /><br />
    <label
class="form_label"for="email"><b>Username</b></label><br><br>
    <input class="form_input"type="text" name= "username"
/><br><br>
    <label class="form_label"for="psw"><b>Pin</b></label><br><br>
    <input class="form_input"type="password" name="pin"/>
    <br></br></br>
    <center><input type="submit" class="submitbtn"value="submit"
/></center>
    </form>
    </div>
    <center><p>Back to <a
href="{{url_for('login')}}">login</a></p></center>
</body>

```

</html>

home.html

```
<!DOCTYPE html>
```

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

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>SKYS.com</title>
```

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

```
  <script>
```

```
    window.watsonAssistantChatOptions = {
```

```
      integrationID: "5bea080a-0793-42c5-97b8-3ede9c7b6a0c", // The  
ID of this integration.
```

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

```
      serviceInstanceID: "ead2bc3a-7163-45a7-ae03-bd16c50d4ecc", //  
The ID of your service instance.
```

```
      onLoad: function (instance) { instance.render(); }
```

```
    };
```

```
    setTimeout(function () {
```

```
      const t = document.createElement('script');
```

```
      t.src = "https://web-
```

```
chat.global.assistant.watson.appdomain.cloud/versions/" +
```

```
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
```

```
"/WatsonAssistantChatEntry.js";
```

```
      document.head.appendChild(t);
```

```
    });
```

```

</script>
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('home')}}">Home</a></li>
        <li><a href="{{url_for('register')}}">Register</a></li>
        <li><a href="{{url_for('login')}}">Login</a></li>
        <li><a href="{{url_for('welcome_page')}}">Apply Job</a></li>
        <li><a href="{{url_for('skills')}}">Skills</a></li>
        <li><a href="{{url_for('about')}}">About us</a></li>
        <li><a href="{{url_for('contact')}}">contact us</a></li>
      </ul>
    </nav>
  </header>
  <section id="sss1">
    <h1 class="two" ><span style="color:rgb(20, 20, 70);background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">SKYS.com</span></h1>
    <div id="ss1">
      <input class="input_search" type="text" placeholder="Search Jobs"
    >
      <button class="button_search">Search</button>
    </div>
    
  </section>

```

```
</body>
</html>
```

login.html

```
<html>
  <head>
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
      <nav class="navbar">
        <ul>
          <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
        </ul>
      </nav>
    </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(109, 60,
225);background-color: whitesmoke;margin-left: 30px;padding-left:
20px;padding-right: 20px;">SKYS</span>.com</h1></center>
    </section>
    <br><br>
    <div class="container"> <br /><br />
    <form action="/login" method="POST">
      <h1>Login</h1> <br /><br />
```

```
<label
class="form_label"for="email"><b>Username</b></label><br><br>
  <input class="form_input"type="text" name= "username"
/><br><br>
  <label
class="form_label"for="psw"><b>Password</b></label><br><br>
  <input class="form_input"type="password" name="password"/>
</br></br></br>
  <center><input type="submit" class="submitbtn"value="submit"
/></center>
</form>
</div>
  <center><p>Forgot your password? <a href="{{url_for('forget')}}">Try
Another Way</a></p></center>
  <center><p>Don't have a account <a
href="{{url_for('register')}}">Create new account</a></p></center>
</body>
</html>
```


GITHUB 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-42576-1660668622>

PROJECT DEMO LINK:

<https://www.youtube.com/embed/LghHoC7KZuA>