Final report

Skill and job recommender application

Mentors:

1)Krishna Chaitanya

2)N.J.Divya

Team ID : PNT2022TMID33163

Team Leader :

1)Ruban M(922119104303)

Team members:

2)Ramji K(922119104033)

3)Sanjay Narayanan S(922119104037)

4)Vasanthan M P(922119104047)

-

CHAPTER **TITLE**

NO

ABSTRACT

Skill / Job Recommender Application Having lots of skills but wondering which job will best suit you? Don’t need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job.
To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

CHAPTER 1

1.INTRODUCTION

Now a days Searching a job through the internet is a common task using various platforms such as linked in, websites and other social media applications such as Instagram. Basically a job seeker can search for a job either by searching for job vacancies that suits his/her profile or by creating a profile stating his/her educational details, professional skill and personal experiences and receive job details based on the provided data. Job recommender systems have implemented many type of recommenders such as content-based filtering, collaborative filtering, knowledge based and hybrid approaches.

1.1 PROJECT OVERVIEW:

Now a days its very difficult for people to find a job in today's competitive world, but skill and job recommender application is used to recommend various types of available vacant job roles to the job seekers. Additionally, it describes the various career paths open to professionals in all fields as well as the various business models used by those employers. Our application recommends job based on the skillset provided by the user.

1.2 PURPOSE

Enormous amounts of jobs are posted on the job websites on daily basis and large numbers of new resumes are also added to job websites daily. In such scenario it’s a very tough job to suggest matching jobs to the job applicants. A recommendation system can solve this problem to the great extent. A recommendation system has
already been proved to be very effective in the area of Online shopping websites and Movie recommendation. Given a user, the goal of an employment recommendation system is to predict those job positions that are likely to be relevant to the user. An Employment recommendation system would suggest matching jobs to the users using matching, collaborative filtering and content based recommendation based on ranking.

CHAPTER 2

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

A lot of research has been carried out in the field of job recommender systems. A large variety of job recommendation systems already exist that try to provide one or the other aspect of the information by applying different methods.The key problem is that most of job hunting websites just provides recruitment information to website viewers. Students have to retrieve information among those displayed by websites to find jobs they want to apply. The whole procedure is lengthy and inefficient. In addition, many ecommerce websites, uses collaborative filtering algorithm without considering user’s resume and item’s properties.

An online job recommendation system that classifies users into groups by using historical behaviors of users and individual information and then uses the appropriate recommendation approach for each group of users. This approach is suitable for the cases in which different users may have different attributes and a single recommendation approach may not be appropriate for all users.

2.2 REFERENCES

1. Dynamic User Profile-Based Job Recommender System - Wenxing Hong, Siting Zheng, Huan Wang School of Information Science and Technology Xiamen University Xiamen, China. The 8th International Conference on Computer Science & Education (ICCSE 2013) April 26-28, 2013. Colombo, Sri Lanka.

2. Kuan Liu, Xing Shi, Anoop Kumar, Linhong Zhu, and Prem Natarajan. 2016. Temporal learning and sequence modeling for a job recommender system. In Proceedings of the Recommender Systems Challenge (RecSys Challenge '16). Association for Computing Machinery, New York, NY, USA, Article 7, 1–4. https://doi.org/10.1145/2987538.2987540

3. Choudhary, S., Koul, S., Mishra, S., Thakur, A., & Jain, R. (2016, October). Collaborative job prediction based on Naïve Bayes Classifier using python platform. In 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS) (pp. 302-306). IEEE.

4. Yang, S., Korayem, M., AlJadda, K., Grainger, T., & Natarajan, S. (2017). Combining contentbased and collaborative filtering for job recommendation system: A cost-sensitive Statistical Relational Learning approach. Knowledge-Based Systems, 136, 37-45.

5. Shalaby, W., AlAila, B., Korayem, M., Pournajaf, L., AlJadda, K., Quinn, S., & Zadrozny, W. (2017, December). Help me find a job:

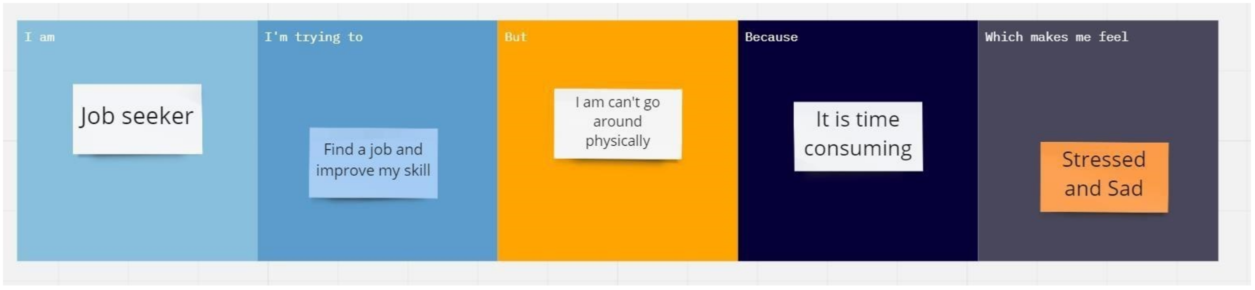
A graph-based approach for job recommendation at scale. In 2017 IEEE international conference on big data (big data) (pp. 1544-1553). IEEE.

6. Dave, V. S., Zhang, B., Al Hasan, M., AlJadda, K., & Korayem, M. (2018, October). A combined representation learning approach for better job and skill recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (pp. 1997-2005).

7. Jorge Valverde-Rebaza, Ricardo Puma, Paul Bustios, Nathalia C Silva - "Job Recommendation based on Job Seeker Skills: An Empirical Study". Conference: March 2018- First Workshop on Narrative Extraction From Text (Text2Story 2018) co-located with 40th European Conference on Information Retrieval (ECIR 2018)At: Grenoble, France.

2.3 PROBLEM STATEMENT DEFINITION

A job seeker always spends hours searching through the massive amount of recruiting information on the internet to identify ones that are helpful. People who don't have any industry experience frequently don't know exactly what they need to learn in order to get a career that's right for them. We deal with the issue of suggesting suitable occupations to those looking for new employment. The goal of job recommender technology is to assist job seekers in Locating the positions that fit their skill set.



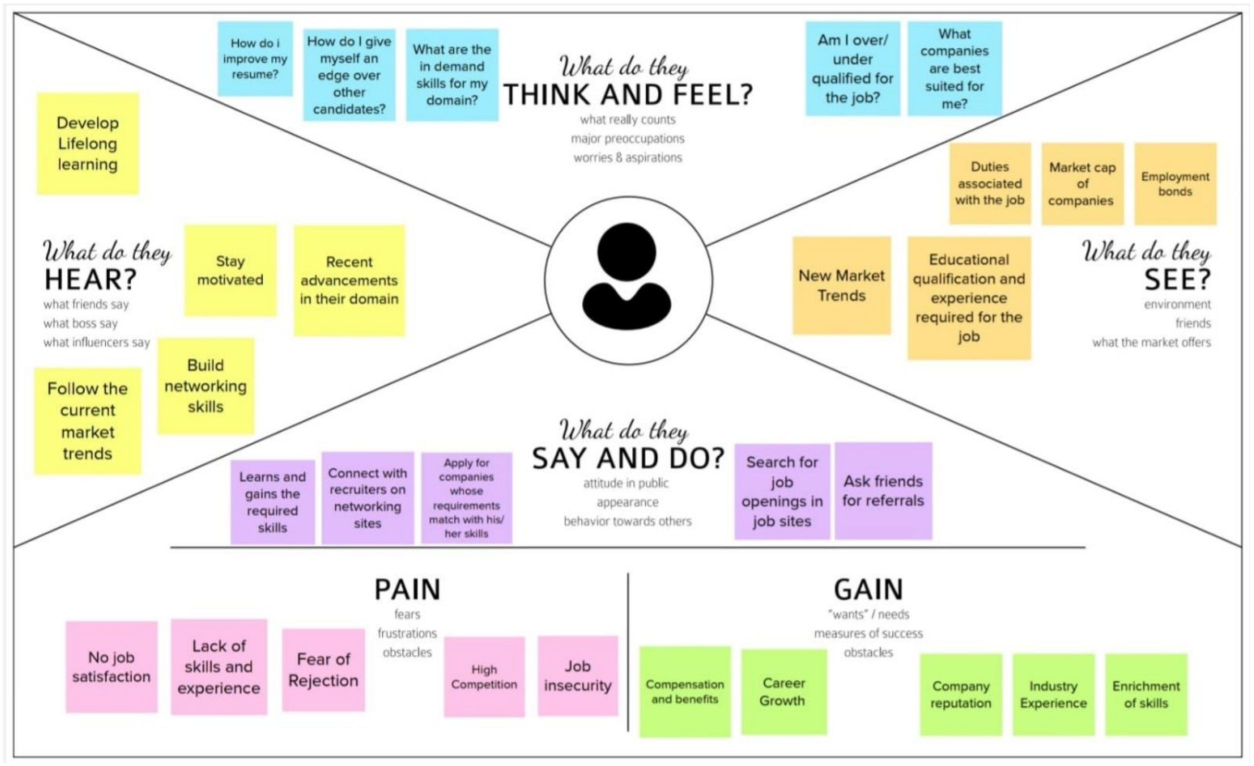
CHAPTER 3

IDEATION& PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes.

It is a useful tool to helps teams better understand their users.
Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

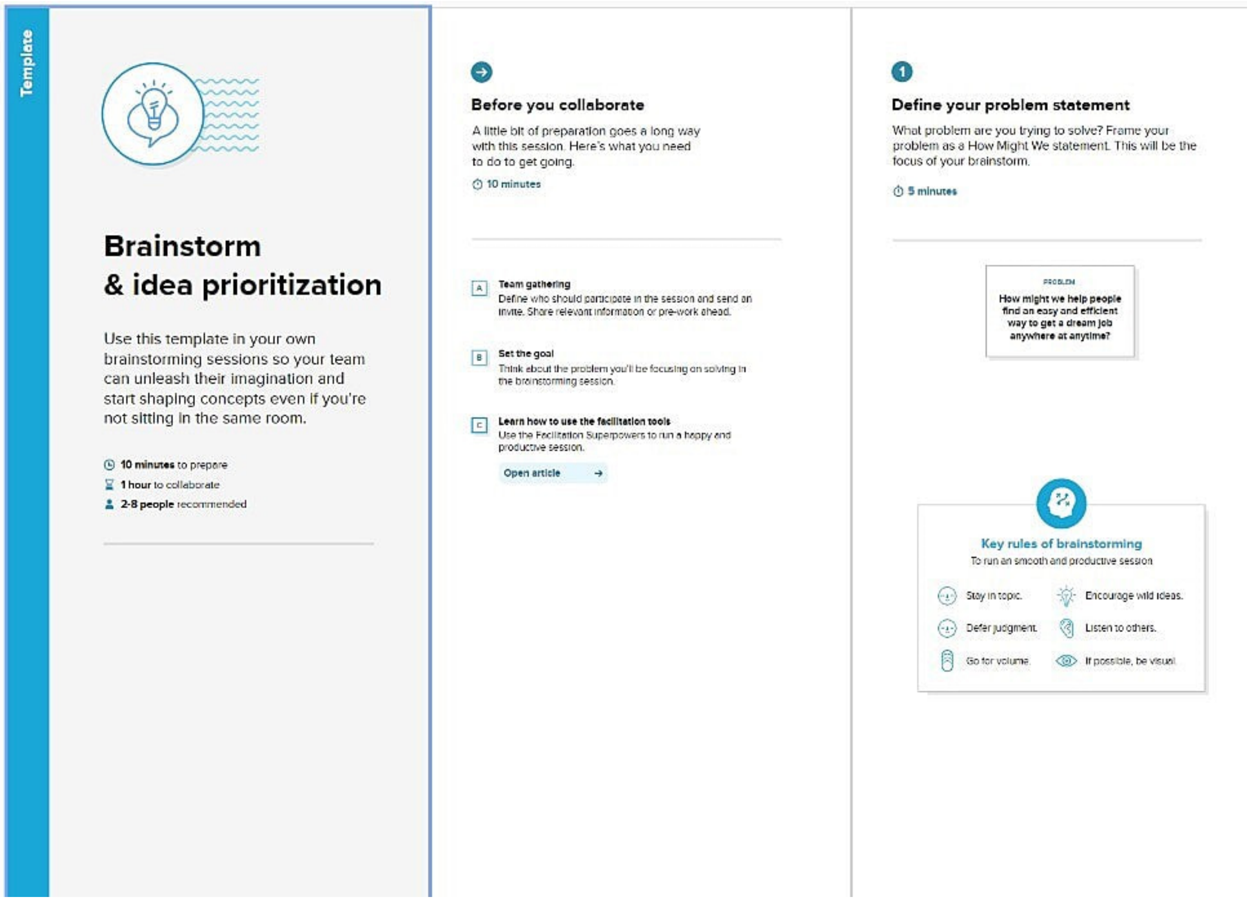


How do i improve my resume?
How do I give myself an edge over other candidates?
What are the in demand skills for my domain?
What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
Am I over/ under qualified for the job?
What companies are best suited for me?
Develop Lifelong learning
Duties associated with the job
Market cap of companies
Employment bonds
What do they HEAR?
what friends say
what boss say
what influencers say
Stay motivated
Recent advancements in their domain
New Market Trends
Educational qualification and experience required for the job
What do they SEE?
environment
friends
what the market offers
Build networking skills
Follow the current market trends
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
Learns and gains the required skills
Connect with recruiters on networking sites
Apply for companies whose requirements match with his/ her skills
Search for job openings in job sites
Ask friends for referrals
PAIN
fears
frustrations
obstacles
No job satisfaction
Lack of skills and experience
Fear of Rejection
High Competition
Job insecurity
GAIN
"wants" / needs
measures of success
obstacles
Compensation and benefits
Career Growth
Company reputation
Industry Experience
Enrichment of skills

3.2 IDEATION AND BRAINSTORMING

Brainstorming is a method design teams use to generate ideas to solve clearly defined design problems. Brainstorming is a method of generating ideas and sharing knowledge to solve a particular commercial or technical problem, in which participants are encouraged to think without interruption. Brainstorming is a group activity where each participant shares their ideas as soon as they come to mind. At the conclusion of the session, ideas are categorised and ranked for follow-on action.

When planning a brainstorming session it is important to define clearly the topic to be addressed. A topic which is too specific can constrict thinking, while an ill-defined topic will not generate enough directly applicable ideas. The composition of the brainstorming group is important too. It should include people linked directly with the subject as well as those who can contribute novel and unexpected ideas. It can comprise staff from inside or outside the organisation.



Template

Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- **10 minutes** to prepare
- **1 hour** to collaborate
- **2-8 people** recommended

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

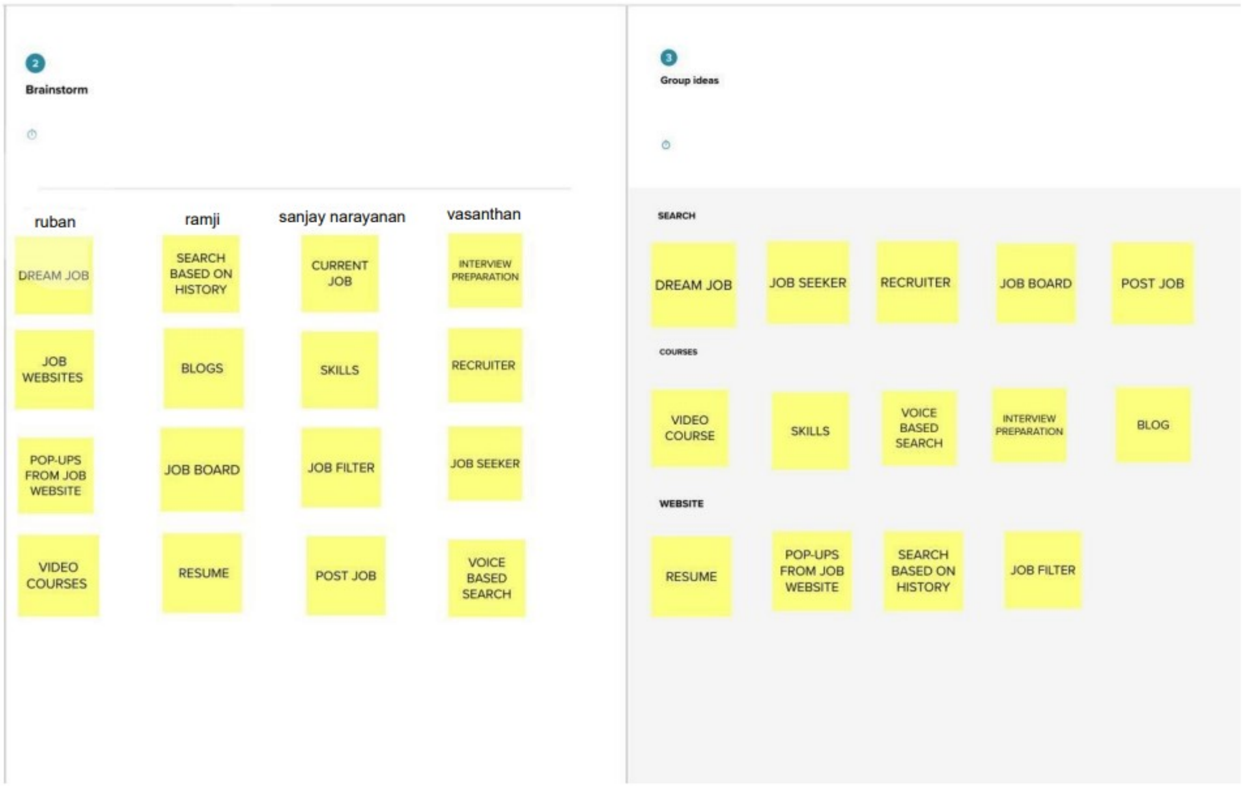
PROBLEM

How might we help people find an easy and efficient way to get a dream job anywhere at anytime?

Key rules of brainstorming

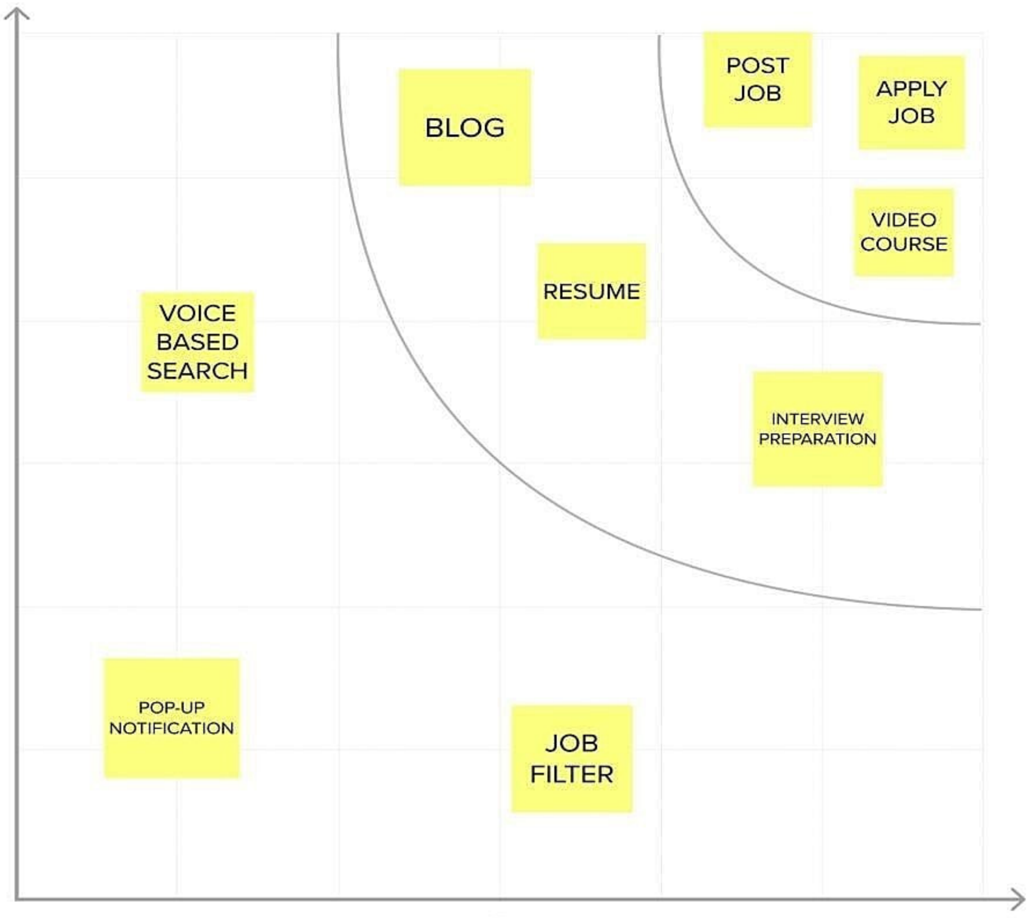
To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.


2
Brainstorm
ruban
ramji
sanjay narayanan
vasanthan
DREAM JOB
SEARCH BASED ON HISTORY
CURRENT JOB
INTERVIEW PREPARATION
JOB WEBSITES
BLOGS
SKILLS
RECRUITER
POP-UPS FROM JOB WEBSITE
JOB BOARD
JOB FILTER
JOB SEEKER
VIDEO COURSES
RESUME
POST JOB
VOICE BASED SEARCH
3
Group ideas
SEARCH
DREAM JOB
JOB SEEKER
RECRUITER
JOB BOARD
POST JOB
COURSES
VIDEO COURSE
SKILLS
VOICE BASED SEARCH
INTERVIEW PREPARATION
BLOG
WEBSITE
RESUME
POP-UPS FROM JOB WEBSITE
SEARCH BASED ON HISTORY
JOB FILTER

4

Prioritize



Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

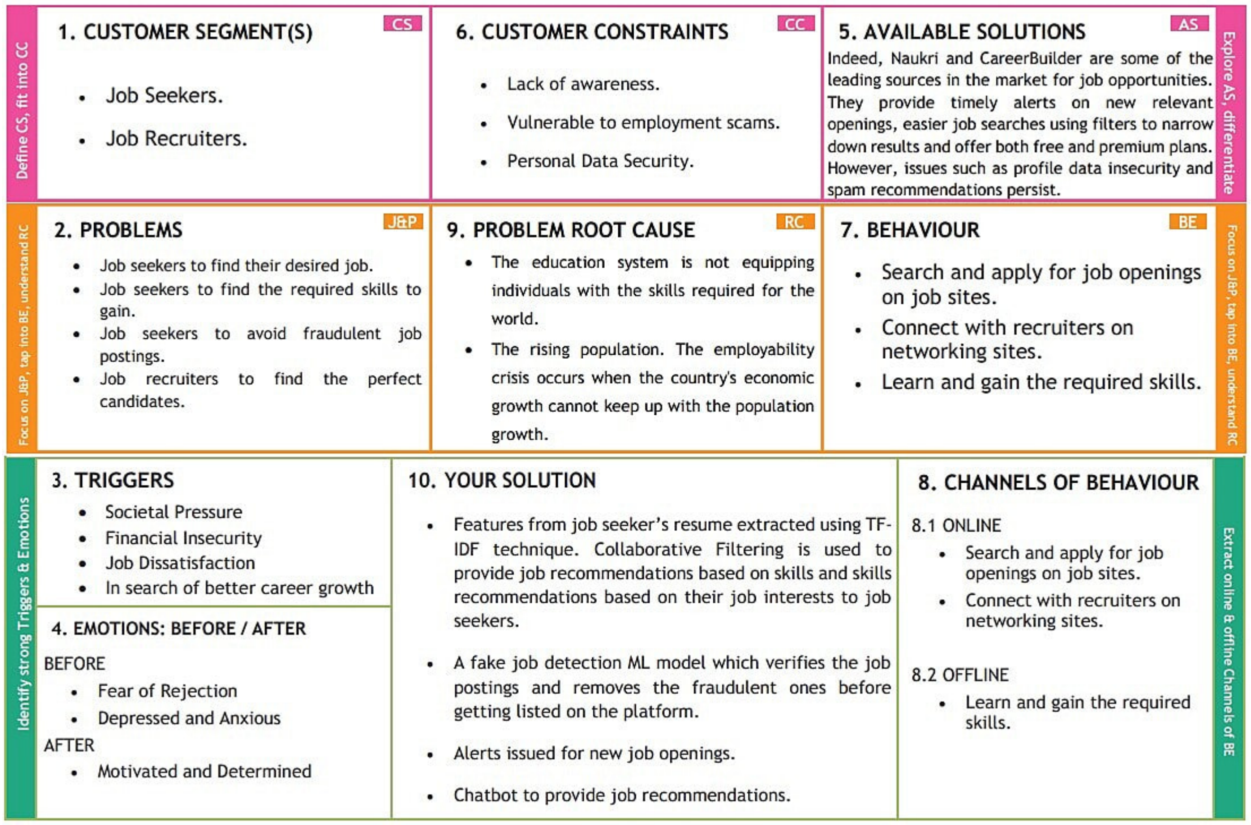
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Unemployment gives rise to poverty, causing a decrease in production and less consumption of goods and services, contributing to the nation's economic loss. Every industry has a lot of career opportunities, but job seekers are unaware of them. The unemployability crisis can be solved if every job seeker receives the right career guidance and proper job role training. On the other hand, recruiters are finding a way to make the hiring procedure easier for choosing potential candidates. Job recruiters also search for a medium to reach out to many job seekers to promote their firm's name. So, to eradicate the unemployment crisis, for the job seekers to find a job they desire, match their qualifications and skills, train themselves for their expected job roles and help the job recruiters find the perfect candidates, we need to develop a skill and job recommendation engine.

2.	Idea / Solution description	The skills (basic features) are extracted from the job seeker's resume using the TF-IDF technique. The job seeker's profile may get outdated sometimes as they fail to update the resume regularly. The dynamic behaviour of the job seeker is noted by observing the jobs he applied for. So, the dynamic features are extracted, which are an updated version of basic features, by making a statistic at regular intervals. The dynamic recommendation engine works as follows: A collaborative user-based filtering algorithm is used initially to overcome the cold-start problem. It takes the features extracted from the job seeker's profile and the features extracted from the job description, computes the similarity between the two using Euclidean distance, and recommends the top k similar jobs applied to generate the initial recommendation jobs. The system provides the initial recommendation to the job seeker and records his behaviour. Thus, we will be able to arrive at a set of jobs in which the job seeker is interested and a set of jobs in which he is not interested. The extended new basic features help in updating the job seeker's profile. Thus, the job applicant is provided with new recommendations. Similarly, the same recommendation system helps provide job applicant recommendations to the job recruiters to find the most eligible candidates for their firm. Training programmes and certification courses are also recommended to job seekers based on their job interests to grow their skills.

3.	Novelty / Uniqueness	A fake job detection ML model which verifies the job postings and removes the fraudulent ones before getting listed on the platform is integrated with the recommendation engine to bring down the employment scams.

4.	Social Impact / Customer Satisfaction	The job and skill recommender system is expected to reduce unemployment and improve the skills of job seekers to boost the country’s economy. The customer satisfaction can be measured by customer loyalty and customer reviews after deployment of the project.
5.	Business Model (Revenue Model)	A subscription model can be provided for both employees and employers with additional costs for features along with recurring monthly or yearly costs.
6.	Scalability of the Solution	In order to provide the best scalability, cloud computing is utilised. The cloud is capable of increasing or decreasing IT resources as needed to meet the changing demand and workload.

3.4 PROBLEM SOLUTION FIT



Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Identify strong Triggers & Emotions
Extract online & offline Channels of BE
CS
CC
AS
J&P
RC
BE

1. CUSTOMER SEGMENT(S)

- Job Seekers.
- Job Recruiters.

6. CUSTOMER CONSTRAINTS

- Lack of awareness.
- Vulnerable to employment scams.
- Personal Data Security.

5. AVAILABLE SOLUTIONS

Indeed, Naukri and CareerBuilder are some of the leading sources in the market for job opportunities. They provide timely alerts on new relevant openings, easier job searches using filters to narrow down results and offer both free and premium plans. However, issues such as profile data insecurity and spam recommendations persist.

2. PROBLEMS

- Job seekers to find their desired job.
- Job seekers to find the required skills to gain.
- Job seekers to avoid fraudulent job postings.
- Job recruiters to find the perfect candidates.

9. PROBLEM ROOT CAUSE

- The education system is not equipping individuals with the skills required for the world.
- The rising population. The employability crisis occurs when the country's economic growth cannot keep up with the population growth.

7. BEHAVIOUR

- Search and apply for job openings on job sites.
- Connect with recruiters on networking sites.
- Learn and gain the required skills.

3. TRIGGERS

- Societal Pressure
- Financial Insecurity
- Job Dissatisfaction
- In search of better career growth

4. EMOTIONS: BEFORE / AFTER

BEFORE
- Fear of Rejection
- Depressed and Anxious

AFTER
- Motivated and Determined

10. YOUR SOLUTION

- Features from job seeker's resume extracted using TF-IDF technique. Collaborative Filtering is used to provide job recommendations based on skills and skills recommendations based on their job interests to job seekers.
- A fake job detection ML model which verifies the job postings and removes the fraudulent ones before getting listed on the platform.
- Alerts issued for new job openings.
- Chatbot to provide job recommendations.

8. CHANNELS OF BEHAVIOUR

8.1 ONLINE
- Search and apply for job openings on job sites.
- Connect with recruiters on networking sites.

8.2 OFFLINE
- Learn and gain the required skills.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	**Functional Requirement (Epic)**	**Sub Requirement (Story / Sub-Task)**
FR-1	User Registration	Registration through Form Registration through Google Account Registration through LinkedIn Account
FR-2	User Login	Login using login credentials Sign in with Google/ LinkedIn Account
FR-3	Search jobs	Job seekers search for jobs by desired role, salary and location.
FR-4	Get Appropriate Job Recommendations	Resume extraction/resume parsing is done to extract useful information from the CV uploaded by the job seekers. The extracted features are the basic features, and the job seeker's activity is recorded, which gives the dynamic features. The basic and dynamic features help provide personalized job recommendations to job seekers via

		chatbot.

FR-5	Get Job Alerts	Timely reminders are provided regarding the deadlines of the application process and new job openings.
FR-6	Get Accurate Skill Recommendations	Job seekers are recommended technical certification courses based on their resumes to become skilled and industry ready.
FR-7	Display Job-Skill Match Score	Job seekers are provided with a match score which indicates how much their skills match the job profile.
FR-8	Post job vacancies	Job recruiters post details of job vacancies in their company. The details include the job description, required qualifications, job responsibilities, working conditions, etc.

FR-9	Fake Job Detection and Removal	The portal has a fake job offer detection engine embedded in it.
FR-10	Bookmark Job Posts	The job seeker can bookmark or tag any number of jobs that they apply for, for quick access.

4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	**Non-Functional Requirement**	**Description**
NFR-1	**Usability**	Intuitive UI is provided to the users, ensuring they can easily navigate through the application.

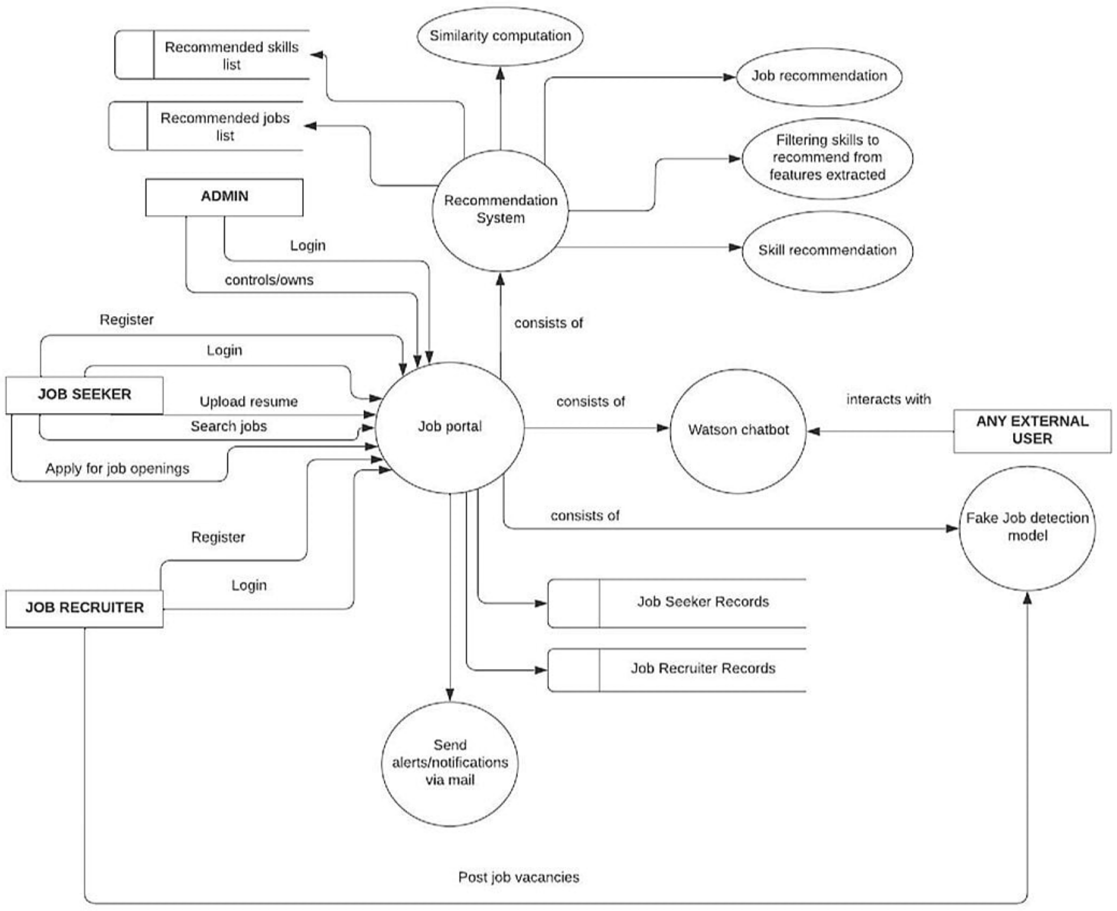
NFR-2	**Security**	**Authentication -** The user must be logged in to view the job posts and apply. **Withholding of sensitive information -** Passwords are not stored within the system or revealed to the users. **Privacy -** The users can choose who can view their profile/posts.
NFR-3	**Reliability**	The system performs without failure in 95% of the use cases. The total downtime for the system over a year shall not exceed 50 hours.
NFR-4	**Performance**	**Processing time -** The processing time takes less than three seconds. **Response time -** The server responds to the client’s requests in less than one second. **Querying time -** Querying the database takes less than one second.

NFR-5	**Availability**	The system is always available.
NFR-6	**Scalability**	To meet the changing demand and workload, cloud services are utilized.
NFR-7	**Recovery**	The system frequently backs up the user data to avoid any data loss. In the event of any disaster, the latest backup is immediately restored.

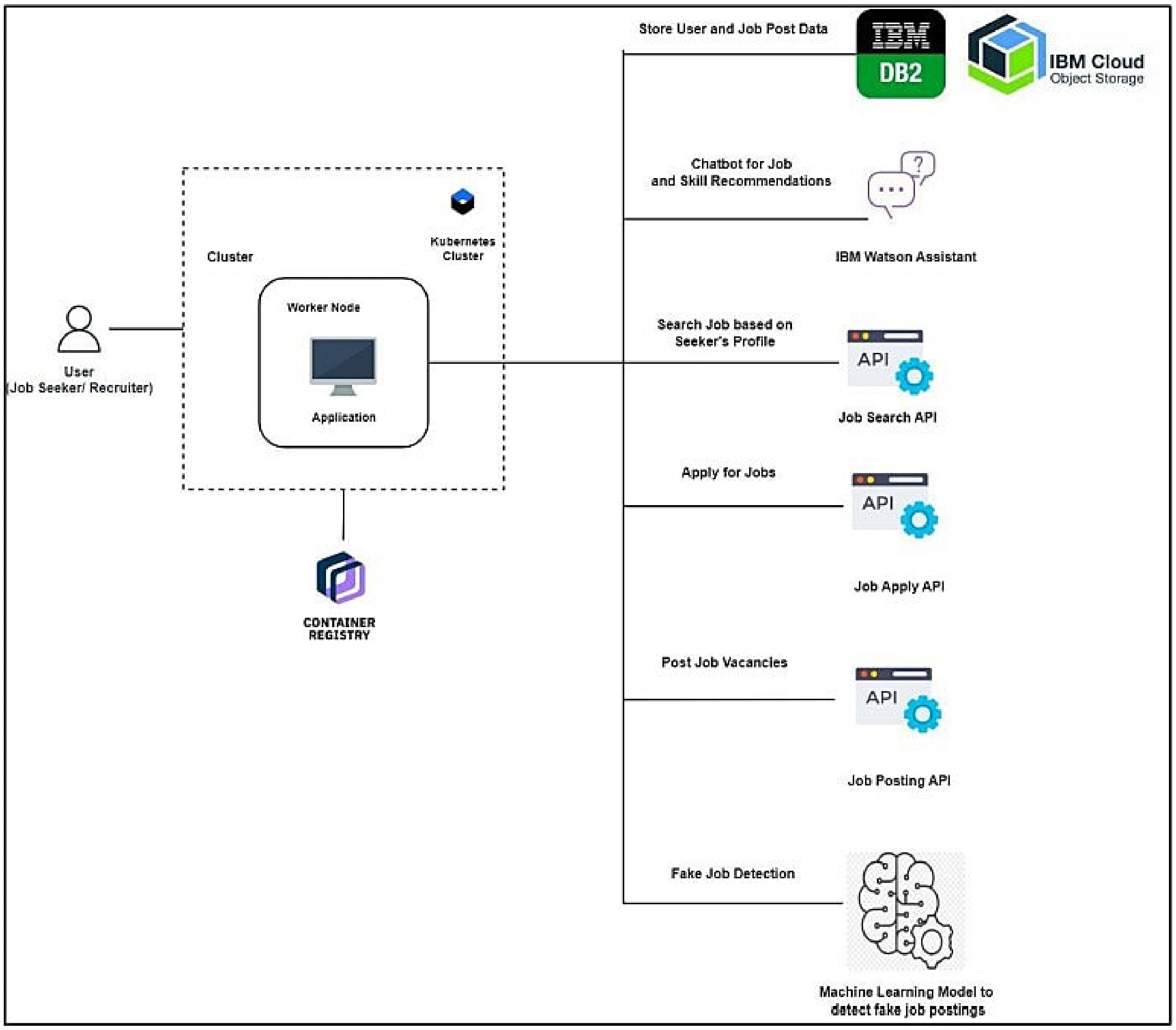
CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE



5.3 USER STORIES

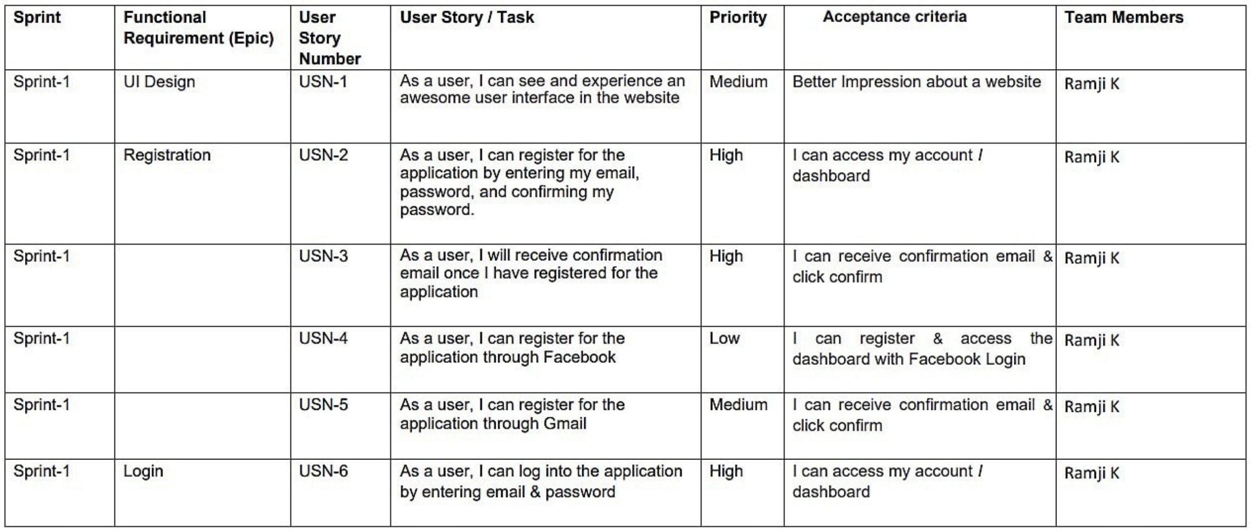
User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance Criteria**	**Priority**	**Release**
Job Seeker/ Recruiter	Registration	USN-1	As a job seeker/ recruiter, I can register for the application by entering my details.	I can access my account/dash board.	High	Sprint 1
		USN-2	As a job seeker/ recruiter, I can register for the application through Google Account.	I can access my account/dash board.	Low	Sprint 1
		USN-3	As a job seeker/ recruiter, I can register for the application through LinkedIn.	I can access my account/dash board.	Low	Sprint 1
Job Seeker/ Recruiter/ Admin	Login	USN-4	As a job seeker/ recruiter, I can log into the application by entering login credentials.	I can access my account/dash board.	High	Sprint 1
Job Seeker	Job Search	USN-5	As a job seeker, I can search for job postings based on preferred job	I can find job postings related to my search.	High	Sprint 1

			roles, location, and salary.			
	Apply for Jobs	USN-6	As a job seeker, I can apply for the jobs listed in the portal.	I can apply for job openings.	High	Sprint 1
	Get Job Recommendations	USN-7	As a job seeker, I upload my resume.	I can get skill and job recommendations by resume parsing.	High	Sprint 2

CHAPTER 6

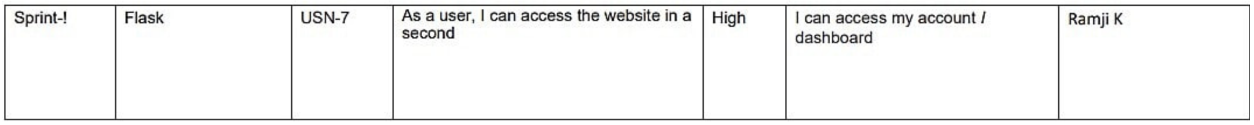
6.PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

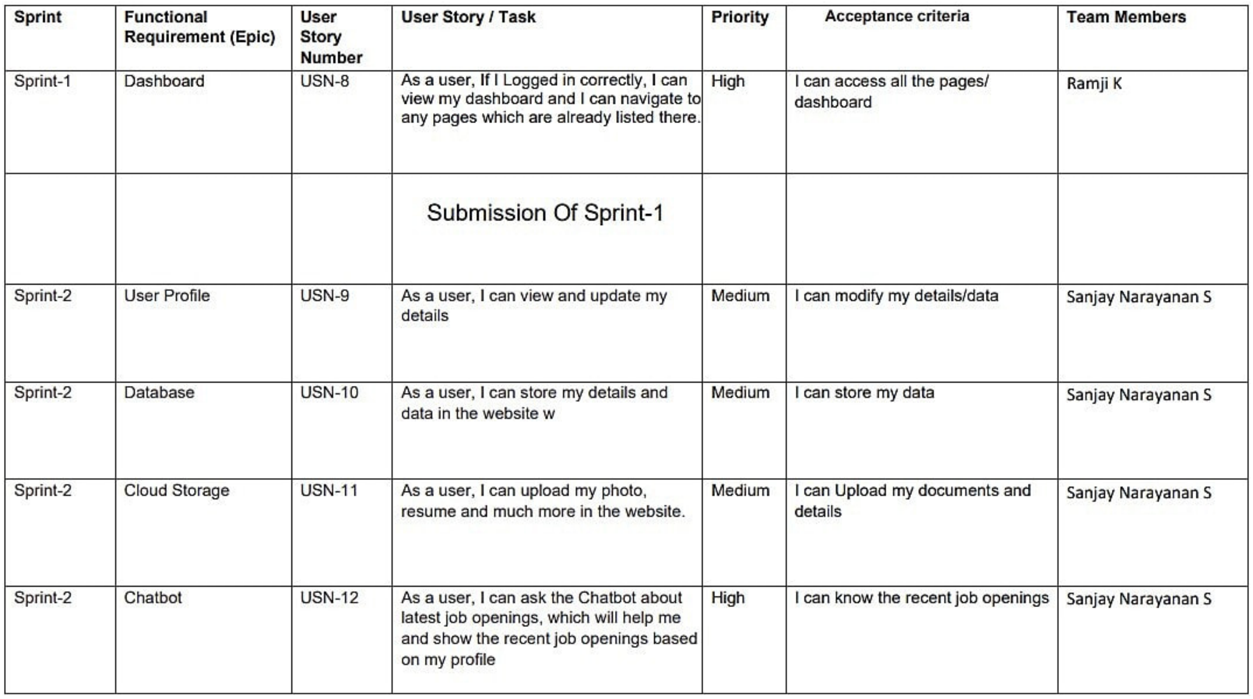


Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	UI Design	USN-1	As a user, I can see and experience an awesome user interface in the website	Medium	Better Impression about a website	Ramji K
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	High	I can access my account / dashboard	Ramji K
Sprint-1		USN-3	As a user, I will receive confirmation email once I have registered for the application	High	I can receive confirmation email & click confirm	Ramji K
Sprint-1		USN-4	As a user, I can register for the application through Facebook	Low	I can register & access the dashboard with Facebook Login	Ramji K
Sprint-1		USN-5	As a user, I can register for the application through Gmail	Medium	I can receive confirmation email & click confirm	Ramji K
Sprint-1	Login	USN-6	As a user, I can log into the application by entering email & password	High	I can access my account / dashboard	Ramji K

6.2 SPRINT DELEVERY SCHEDULE



Sprint-!	Flask	USN-7	As a user, I can access the website in a second	High	I can access my account / dashboard	Ramji K



Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	Dashboard	USN-8	As a user, If I Logged in correctly, I can view my dashboard and I can navigate to any pages which are already listed there.	High	I can access all the pages/ dashboard	Ramji K
			Submission Of Sprint-1			
Sprint-2	User Profile	USN-9	As a user, I can view and update my details	Medium	I can modify my details/data	Sanjay Narayanan S
Sprint-2	Database	USN-10	As a user, I can store my details and data in the website w	Medium	I can store my data	Sanjay Narayanan S
Sprint-2	Cloud Storage	USN-11	As a user, I can upload my photo, resume and much more in the website.	Medium	I can Upload my documents and details	Sanjay Narayanan S
Sprint-2	Chatbot	USN-12	As a user, I can ask the Chatbot about latest job openings, which will help me and show the recent job openings based on my profile	High	I can know the recent job openings	Sanjay Narayanan S

Sprint-2	Identity-Aware	USN-13	As a User, I can access my account by entering by correct login credentials. My user credentials is only displayed to me.	High	I can have my account safely	Sanjay Narayanan S
			Submission of Sprint-2			

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-3	Sendgrid service	USN-14	As a user, I can get a notification or mail about a job opening with the help of sendgrid service.	Medium	I can get a notification in a second.	Vasanthan M P
Sprint-3	Learning Resource	USN-15	As a user, I can learn the course and I will attain the skills which will be useful for developing my technical skills.	High	I can gain the knowledge and skills	Vasanthan M P
Sprint-3	Docker	USN-16	As a user, I can access the website in any device	High	I can access my account in any device	Vasanthan M P
Sprint-3	Kubernates	USN-17	As a user, I can access the website in any device	High	I can access my account in any device	Vasanthan M P
Sprint-3	Deployment in cloud	USN-18	As a user, I can access the website in any device	High	I can access my account in any device	Vasanthan M P
Sprint-3	Technical support	USN-19	As a user, I can get a customer care support from the website which will solve my queries.	Medium	I can tackle my problem & queries.	Vasanthan M P

			Submission of Sprint-3			
Sprint-4	Unit Testing	USN-15	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Ruban M
Sprint-4	Integration testing	USN-16	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Ruban M

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-4	System testing	USN-17	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Ruban M
Sprint-4	Correction	USN-18	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Ruban M
Sprint-4	Acceptance testing	USN-19	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Ruban M
			Submission of Sprint-4			

CHAPTER 7

CODING AND SOLUTIONING

app.py

```
import re
from flask import Flask, render_template, request, redirect, url_for, session
import ibm_db
from flask_mail import Mail, Message


app = Flask(__name__)


app.secret_key = "a"
conn = ibm_db.connect(
    "DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=czt98936;PWD=DICkH4DQDETaI9Hm",
    "",
    "",
)
```

```
@app.route("/")
def home():
    return render_template("index.html")



@app.route("/login", methods=["GET", "POST"])
def login():
    global userid
    msg = ""

    if request.method == "POST":
        username = request.form["username"]
        password = request.form["password"]
        sql = "SELECT * FROM users 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"]
            userid = account["USERNAME"]
            session["username"] = account["USERNAME"]
            msg = "Logged in succesfully"

            return render_template("ViewJob.html", msg=msg)
        else:
            msg = "Incorrect username/password!"
    return render_template("index.html", msg=msg)



@app.route("/register", methods=["GET", "POST"])
def register():
    msg = ""
    if request.method == "POST":
        username = request.form["username"]
        email = request.form["email"]
        password = request.form["password"]
        sql = "SELECT * FROM users WHERE username=?"
        stmt = ibm_db.prepare(conn, sql)
```

```
ibm_db.bind_param(stmt, 1, username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    msg = "Account Already exists"
elif not re.match(r"[^@]+@[^@]+\.[^@]+", email):
    msg = "Invalid email"
elif not re.match(r"[A-Za-z0-9]+", username):
    msg = "name must contain only alpha characters or numbers!"
else:
    insert_sql = "INSERT INTO users VALUES(?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.bind_param(prep_stmt, 2, email)
    ibm_db.bind_param(prep_stmt, 3, password)
    ibm_db.execute(prep_stmt)
    msg = "you have successfully registered"
    app.config["SECRET_KEY"] = "top-secret!"
    app.config["MAIL_SERVER"] = "smtp.sendgrid.net"
    app.config["MAIL_PORT"] = 587
    app.config["MAIL_USE_TLS"] = True
```

```
            app.config["MAIL_USERNAME"] = "apikey"
            app.config[
                "MAIL_PASSWORD"
            ] = "SG.XbHqaobAQPCL5ZW_3-jRkA.vuaAYWQBDNuRV-
5MjIVERohpOKt-dKvcQmXNsgjFi74"
            app.config["MAIL_DEFAULT_SENDER"] =
"rubantamilboy@gmail.com"
            mail = Mail(app)
            recipient = request.form["email"]
            msg = Message("Successfully Registered",
recipients=[recipient])
            msg.body = (
                "Congratulations! You have successfully registered with "
                "Skill/Job Recommender!"
            )
            msg.html = (
                "<h1>Successfully Registered</h1>"
                "<p>Congratulations! You have successfully registered with "
                "<b>Skill/Job Recommender</b>!</p>"
            )
            mail.send(msg)
            return render_template("index.html")
    elif request.method == "POST":
        msg = "Please fill out the form!"
    return render_template("index.html", msg=msg)



@app.route("/view")
def view():
    return render_template("ViewJob.html")



@app.route("/apply", methods=["GET", "POST"])
def apply():
    msg = ""
    if request.method == "POST":
        username = request.form["username"]
        email = request.form["email"]
        qualification = request.form["qualification"]
        skills = request.form["skills"]
        jobs = request.form["jobs"]

        insert_sql = "INSERT INTO jobs values(?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, username)
        ibm_db.bind_param(prep_stmt, 2, email)
        ibm_db.bind_param(prep_stmt, 3, qualification)
        ibm_db.bind_param(prep_stmt, 4, skills)
        ibm_db.bind_param(prep_stmt, 5, jobs)
        ibm_db.execute(prep_stmt)
        msg = "You have succesfully applied for the job!"
        session["loggedin"] = True
        app.config["SECRET_KEY"] = "top-secret!"
        app.config["MAIL_SERVER"] = "smtp.sendgrid.net"
        app.config["MAIL_PORT"] = 587
        app.config["MAIL_USE_TLS"] = True
        app.config["MAIL_USERNAME"] = "apikey"
        app.config[
            "MAIL_PASSWORD"
        ] = "SG.XbHqaobAQPCL5ZW_3-jRkA.vuaAYWQBDNuRV-
5MjIVERohpOKt-dKvcQmXNsgjFi74"
        app.config["MAIL_DEFAULT_SENDER"] =
"rubantamilboy@gmail.com"
        mail = Mail(app)
        recipient = request.form["email"]
        msg = Message("Successfully Applied", recipients=[recipient])
        msg.body = (
            "Congratulations! You have successfully applied your job with "
            "Skill/Job Recommender!"
        )
        msg.html = (
            "<h1>Successfully Applied</h1>"
            "<p>Congratulations! You have successfully applied your job
with "
            "<b>Skill/Job Recommender</b>!</p>"
        )
        mail.send(msg)
        return redirect(url_for("login"))


    elif request.method == "POST":
        msg = "Please fill the form!"
    return render_template("ApplyJob.html", msg=msg)




@app.route("/logout")
def logout():
    session.pop("loggedin", None)
    session.pop("id", None)
    session.pop("username", None)
    return render_template("index.html")
```

index.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>Registration and Login</title>
        <script defer
src="https://use.fontawesome.com/releases/v5.15.4/js/all.js"
                integrity="sha384-
rOA1PnstxnOBLzCLMcre8ybwbTmemjzdNlILg8O7z1lUkLXozs4DHo
nlDtnE7fpc"
                crossorigin="anonymous"></script>
        <style>
                @import
url('https://fonts.googleapis.com/css?family=Montserrat:400,800');


                * {
```

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

body {
	background: #f6f5f7;
	display: flex;
	justify-content: center;
	align-items: center;
	flex-direction: column;
	font-family: 'Montserrat', sans-serif;
	height: 100vh;
	margin: -20px 0 50px;
}

h1 {
	font-weight: bold;
	margin: 0;
}

h2 {
	text-align: center;
```

```
}

p {
	font-size: 14px;
	font-weight: 100;
	line-height: 20px;
	letter-spacing: 0.5px;
	margin: 20px 0 30px;
}

span {
	font-size: 12px;
}

a {
	color: #333;
	font-size: 14px;
	text-decoration: none;
	margin: 15px 0;
}
```

```
button {
	border-radius: 20px;
	border: 1px solid #41d3ff;
	background-color: #41d3ff;
	color: #FFFFFF;
	font-size: 12px;
	font-weight: bold;
	padding: 12px 45px;
	letter-spacing: 1px;
	text-transform: uppercase;
	transition: transform 80ms ease-in;
}

button:active {
	transform: scale(0.95);
}

button:focus {
	outline: none;
}
```

```
button.ghost {
	background-color: transparent;
	border-color: #FFFFFF;
}

form {
	background-color: #FFFFFF;
	display: flex;
	align-items: center;
	justify-content: center;
	flex-direction: column;
	padding: 0 50px;
	height: 100%;
	text-align: center;
}

input {
	background-color: #eee;
	border: none;
	padding: 12px 15px;
	margin: 8px 0;
	width: 100%;
}

.container {
	background-color: #fff;
	border-radius: 10px;
	box-shadow: 0 14px 28px rgba(0, 0, 0, 0.25),
		0 10px 10px rgba(0, 0, 0, 0.22);
	position: relative;
	overflow: hidden;
	width: 768px;
	max-width: 100%;
	min-height: 480px;
}

.form-container {
	position: absolute;
	top: 0;
	height: 100%;
	transition: all 0.6s ease-in-out;
}
```

```
.sign-in-container {
	left: 0;
	width: 50%;
	z-index: 2;
}

.container.right-panel-active .sign-in-container {
	transform: translateX(100%);
}

.sign-up-container {
	left: 0;
	width: 50%;
	opacity: 0;
	z-index: 1;
}

.container.right-panel-active .sign-up-container {
	transform: translateX(100%);
	opacity: 1;
```

```
        z-index: 5;
        animation: show 0.6s;
}

@keyframes show {

        0%,
        49.99% {
                opacity: 0;
                z-index: 1;
        }

        50%,
        100% {
                opacity: 1;
                z-index: 5;
        }
}

.overlay-container {
        position: absolute;
        top: 0;
        left: 50%;
        width: 50%;
        height: 100%;
        overflow: hidden;
        transition: transform 0.6s ease-in-out;
        z-index: 100;
    }

    .container.right-panel-active .overlay-container {
        transform: translateX(-100%);
    }

    .overlay {
        background: #41d3ff;
        background: -webkit-linear-gradient(to right, #41d3ff,
#FF416C);
        background: linear-gradient(to right, #41d3ff,
#FF416C);
        background-repeat: no-repeat;
        background-size: cover;
        background-position: 0 0;
        color: #FFFFFF;
        position: relative;
        left: -100%;
        height: 100%;
        width: 200%;
        transform: translateX(0);
        transition: transform 0.6s ease-in-out;
}

.container.right-panel-active .overlay {
        transform: translateX(50%);
}

.overlay-panel {
        position: absolute;
        display: flex;
        align-items: center;
        justify-content: center;
        flex-direction: column;
        padding: 0 40px;
        text-align: center;
        top: 0;
        height: 100%;
        width: 50%;
        transform: translateX(0);
        transition: transform 0.6s ease-in-out;
}

.overlay-left {
        transform: translateX(-20%);
}

.container.right-panel-active .overlay-left {
        transform: translateX(0);
}

.overlay-right {
        right: 0;
        transform: translateX(0);
}

.container.right-panel-active .overlay-right {
```

```
                transform: translateX(20%);
            }

            .social-container {
                margin: 20px 0;
            }

            .social-container a {
                border: 1px solid #DDDDDD;
                border-radius: 50%;
                display: inline-flex;
                justify-content: center;
                align-items: center;
                margin: 0 5px;
                height: 40px;
                width: 40px;
            }
        </style>
    </head>

    <body>
```

```
        <h2>Welcome To Skill/Job Recommender Application</h2>
        <div class="container" id="container">
            <div class="form-container sign-up-container">
                <form action="/register" method="post">
                    <h1>Create Account</h1>
                    <span>or use your email for registration</span>
                    <input type="text" placeholder="Name"
    name="username" />
                    <input type="email" placeholder="Email"
    name="email" />
                    <input type="password" placeholder="Password"
    name="password" />
                    <button>Sign Up</button>
                </form>
                <div class="note mt-3 text-center">
                    <!--Register form -->
                    <p> Already have an account ? Please Login <a
    href="/login">login! </a> </p>
                </div>
            </div>
            <div class="form-container sign-in-container">
                <form action="/login" method="post">
                    <div class="msg">{{ msg }}</div>
                    <h1>Sign in</h1>
                    <span>or use your account</span>
                    <input type="text" placeholder="Name"
name="username" />
                    <input type="text" placeholder="Password"
name="password" />
                    <button><a style="color: #fff;">Sign
In</button></a>
                </form>
            </div>
            <div class="overlay-container">
                <div class="overlay">
                    <div class="overlay-panel overlay-left">
                        <h1>Welcome Back!</h1>
                        <p>To keep connected with us please login
with your personal info</p>
                        <button class="ghost" id="signIn">Sign
In</button>
                    </div>
                    <div class="overlay-panel overlay-right">
                        <h1>Hello, Friend!</h1>
                        <p>Enter your personal details and start
journey with us</p>
                        <button class="ghost" id="signUp">Sign
```

```
Up</button>
                        </div>
                    </div>
                </div>
            </div>

            <script>
                const signUpButton = document.getElementById('signUp');
                const signInButton = document.getElementById('signIn');
                const container = document.getElementById('container');

                signUpButton.addEventListener('click', () => {
                    container.classList.add("right-panel-active");
                });

                signInButton.addEventListener('click', () => {
                    container.classList.remove("right-panel-active");
                });
            </script>
</body>
```

```
</html>
```

ViewJob.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>View Job</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.mi
n.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwN
LD69Npy4HI+N" crossorigin="anonymous" />

    <style>
      @import url(https://fonts.googleapis.com/css?family=Hind);
      @import url(https://fonts.googleapis.com/css?family=Open+Sans);

      body {
        font-family: 'Open Sans', sans-serif;
        color: #31383d;
        font-size: 1.1rem;
      }

      section {
        padding: 30px 0;
        background: #fff;
      }

      p {
        line-height: 1.5;
        margin: 30px 0;
      }

      h1,
      h2,
      h3,
      h4,
      h5,
      h6 {
        font-family: 'Hind', sans-serif;
        font-weight: 800;
```

```
}

a {
  color: #31383d;
  -webkit-transition: all 0.2s;
  -moz-transition: all 0.2s;
  transition: all 0.2s;
}

a:hover,
a:focus {
  color: #73a5fc;
}

#navbar-main {
  position: absolute;
  font-family: 'Hind', sans-serif;
  background-color: #fff;
  border-bottom: 1px solid #d7e2e9;
}
```

```
#navbar-main .navbar-brand {
  color: #96a4b1;
}

#navbar-main .navbar-toggler {
  padding: 0.5rem;
  border: none;
}

#navbar-main .navbar-toggler-icon {
  background-image: url("data:image/svg+xml;charset=utf8,%3Csvg
viewBox='0 0 30 30' xmlns='http://www.w3.org/2000/svg'%3E%3Cpath
stroke='rgb(68, 189, 255)' stroke-width='2' stroke-linecap='round'
stroke-miterlimit='10' d='M4 8h24M4 16h24M4
24h24'/%3E%3C/svg%3E");
}

#navbar-main .navbar-nav > li.nav-item > a {
  text-transform: uppercase;
  font-size: 0.875rem;
  font-weight: 600;
  letter-spacing: 1px;
```

```
    text-align: center;
  }

  #navbar-main .navbar-brand .fa-cube {
    font-size: 2rem;
  }

  header.masthead {
    padding-top: 4rem;
    padding-bottom: 4rem;
    margin-bottom: 3rem;
    background: #a3bded;
    background: -webkit-linear-gradient(-20deg, #a3bded 0%, #6991c7
100%);
    background: -moz-linear-gradient(-20deg, #a3bded 0%, #6991c7
100%);
    background: linear-gradient(-20deg, #a3bded 0%, #6991c7 100%);
  }

  header.masthead .site-heading {
    padding: 100px 0 50px;
    color: #fff;
```

```
}

header.masthead .site-heading h1 {
  font-size: 2.3rem;
}

header.masthead .site-heading .subheading {
  display: block;
  font-weight: 300;
  margin: 0.625rem 0 0;
  color: #fff;
}

ul.job-list {
  margin: 0;
  padding: 0;
  list-style: none;
}

ul.job-list > li.job-preview {
  background: #fff;
  border: 1px solid #d7e2e9;
  -webkit-border-radius: 0.4rem;
  -moz-border-radius: 0.4rem;
  border-radius: 0.4rem;
  padding: 1.5rem 2rem;
  margin-bottom: 1rem;
  float: left;
  width: 100%;
  -webkit-transition: all 0.3s ease 0s;
  -moz-transition: all 0.3s ease 0s;
  transition: all 0.3s ease 0s;
}

ul.job-list > li.job-preview:hover {
  cursor: pointer;
  -webkit-box-shadow: 0 3px 8px rgba(0, 0, 0, 0.05);
  -moz-box-shadow: 0 3px 8px rgba(0, 0, 0, 0.05);
  box-shadow: 0 3px 8px rgba(0, 0, 0, 0.05);
}

.job-title {
  margin-top: 0.6rem;
}

.company {
  color: #96a4b1;
}

.job-preview .btn {
  margin-top: 1.1rem;
}

.btn-apply {
  text-transform: uppercase;
  font-size: 0.875rem;
  font-weight: 800;
  letter-spacing: 1px;
  background-color: transparent;
  color: #393a5f;
  border: 2px solid #393a5f;
  padding: 0.6rem 2rem;
  -webkit-border-radius: 2rem;
  -moz-border-radius: 2rem;
  border-radius: 2rem;
}

.btn-apply:hover {
  background-color: #393a5f;
  color: #fff;
  border: 2px solid #393a5f;
}

@media (max-width: 575px) {
  .job-preview .content {
    width: 100%;
  }
}

@media only screen and (min-width: 992px) {
  #navbar-main {
    background: transparent;
    border-bottom: 1px solid transparent;
  }
```

```
#navbar-main .navbar-brand {
  color: #fff;
  opacity: 0.8;
  padding: 0.95rem 1.2rem;
}

#navbar-main .navbar-brand:hover,
#navbar-main .navbar-brand:focus {
  opacity: 1;
}

#navbar-main .navbar-nav > li.nav-item > a {
  color: #fff;
  opacity: 0.8;
  padding: 0.95rem 1.2rem;
}

#navbar-main .navbar-nav > li.nav-item > a:hover,
#navbar-main .navbar-nav > li.nav-item > a:focus {
  opacity: 1;
```

```
        }
      }
    </style>
  </head>

  <body>
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container-fluid">
        <a class="main-logo-img mt-3" href="/registration and
login.html"><img src="https://skilandjobassignment.s3.jp-tok.cloud-
object-storage.appdomain.cloud/undraw_job_hunt_re_q203.svg"
alt="logo" style="border-radius: 10px" width="100px" /> </a>
        <div class="row donate-sponsor">
          <a type="button" class="btn btn-danger mr-1" id="donate"
href="/logout">Logout</a>
        </div>
      </div>
    </nav>
    <header class="masthead">
      <div class="container">
        <div class="row">
          <div class="col-md-10 offset-md-1">
            <div class="site-heading">
```

```
              <h1 class="heading">Open Positions</h1>
              <span class="subheading"> Current listings for jobs. </span>
            </div>
          </div>
        </div>
      </div>
    </header>

    <section>
      <div class="container">
        <div class="row">
          <div class="col-md-10 offset-md-1">
            <ul class="job-list">
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Senior Web Designer</h4>
                  <h5 class="company">Seattle, WA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
              <li class="job-preview">
          <div class="content float-left">
            <h4 class="job-title">Front-End Engineer</h4>
            <h5 class="company">New York, NY</h5>
          </div>
          <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
        </li>
        <li class="job-preview">
          <div class="content float-left">
            <h4 class="job-title">UI/UX Designer</h4>
            <h5 class="company">Los Angeles, CA</h5>
          </div>
          <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
        </li>
        <li class="job-preview">
          <div class="content float-left">
            <h4 class="job-title">Software Developer</h4>
            <h5 class="company">New York, NY</h5>
          </div>
          <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
        </li>
        <li class="job-preview">
          <div class="content float-left">
            <h4 class="job-title">Web Developer</h4>
            <h5 class="company">Los Angeles, CA</h5>
          </div>
          <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
        </li>
        <li class="job-preview">
          <div class="content float-left">
            <h4 class="job-title">Web Designer &amp; Developer</h4>
            <h5 class="company">Seattle, WA</h5>
          </div>
          <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
        </li>
        <li class="job-preview">
          <div class="content float-left">
            <h4 class="job-title">Junior Programmer</h4>
            <h5 class="company">Seattle, WA</h5>
          </div>
          <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
```

```
              </li>
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Visual Designer</h4>
                  <h5 class="company">Los Angeles, CA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Senior Programmer</h4>
                  <h5 class="company">Seattle, WA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </section>
    <script>
      window.watsonAssistantChatOptions = {
        integrationID: '7c500335-c4a6-48a7-820b-80449bc10e0e', // The
ID of this integration.
        region: 'au-syd', // The region your integration is hosted in.
        serviceInstanceID: 'c096c74a-c1fc-422b-b91b-c1f68b210cb5', //
The ID of your service instance.
        onLoad: function (instance) {
          instance.render()
        },
      }
      setTimeout(function () {
        const t = document.createElement('script')
        t.src = 'https://web-
chat.global.assistant.watson.appdomain.cloud/versions/' +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
'/WatsonAssistantChatEntry.js'
        document.head.appendChild(t)
      })
    </script>
  </body>
</html>
```

Applyjob.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>Apply For Job</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.mi
n.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwN
LD69Npy4HI+N" crossorigin="anonymous" />
  </head>

  <body style="padding: 2rem 0; background-color: #eff5f5">
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container-fluid">
        <a class="main-logo-img mt-3" href="/registration and
login.html"><img src="https://skilandjobassignment.s3.jp-tok.cloud-
object-storage.appdomain.cloud/undraw_job_hunt_re_q203.svg"
alt="logo" style="border-radius: 10px; margin-top: 10px"
width="100px" /> </a>
```

```
        <div class="row donate-sponsor">
          <a type="button" class="btn btn-primary mr-1" id="sponsor"
href="/view">View Jobs</a>
          <a type="button" class="btn btn-danger mr-1" id="donate"
href="/logout">Logout</a>
        </div>
      </div>
    </nav>
    <div class="container">
      <div class="msg" style="text-align: center">{{ msg }}</div>
      <h1 style="text-align: center; margin-top: 10px">Apply For
Job</h1>
      <br />
      <div class="row mx-0 justify-content-center">
        <div class="col-md-7 col-lg-5 px-lg-2 col-xl-4 px-xl-0 px-xxl-3"
style="border: 6px solid black; border-radius: 12px">
          <form method="POST" class="w-100 rounded-1 p-4 border bg-
white" action="/apply" enctype="multipart/form-data">
            <label class="d-block mb-4">
              <span class="form-label d-block">Your name</span>
              <input required name="username" type="text" class="form-
control" placeholder="Enter Your Name" />
            </label>
            <label class="d-block mb-4">
              <span class="form-label d-block">Email address</span>
              <input required name="email" type="email" class="form-
control" placeholder="Enter your mail id" />
            </label>
            <label class="d-block mb-4">
              <span class="form-label d-block">Qualification</span>
              <input type="text" name="qualification" class="form-control"
placeholder="Enter your qualification" />
            </label>


            <label class="d-block mb-4">
              <span class="form-label d-block">Skills</span>
              <input type="text" name="skills" class="form-control"
placeholder="Enter your skills" />
            </label>


            <label class="d-block mb-4">
              <span class="form-label d-block">Job Role</span>
              <input type="text" name="jobs" class="form-control"
placeholder="Applying For" />
            </label>
            <div class="mb-3">
              <button type="submit" class="btn btn-primary px-3 rounded-
3">Apply</button>
            </div>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>
```

8. Testing:

8.1 Test cases:

9.Results:

9.1 Performance metrics:

The project has been completed and executed as we expected. All the details were stored in the IBM DB2 and the app was deployed to the cloud. We ensured that the database was well connected using python flask and details were stored. the motive of recommending jobs to the seekers were implemented and was testing by means of various testing methodologies and so expected outcome was obtained.

10.Advantages and disadvantages:

Advantages:

1.Based on their skill set, a person looking for work can quickly locate a position that suits them.

2.A person can take an eligibility exam to determine their eligibility. 3.The majority of recruiters choose the right candidate based on the scores they received on the eligibility tests.

Disadvantages:

1.Job of Person While taking the eligibility test, there can be technical issues.
2.Job seekers may find it difficult to get in touch with recruiters directly.

3.Irrelevant jobs will be suggested if data is not sufficient.

4.The web application will take more time to load if it is facing huge requests.

11.Conclusion:

The application was created to simplify the job hunt. We created an application that is easy to use. based on their skill set, a user can quickly obtain employment. The use of this application benefits the jobseeker without a doubt. Additionally, chatbot is implemented and integrated which provides guidance for the job seekers to apply for the job. 49 12.Future scope: The popular programme linked in allows users to look for work and maintain connections with colleagues and organisations Employers and job seekers both use LinkedIn to locate employment. There are many opportunities to improve our project in the future in a manner similar to linked in.

12.Future scope: The popular programme linked in allows users to look for work and maintain connections with colleagues and organisations Employers and job seekers both use LinkedIn to locate employment. There are many opportunities to improve our project in the future in a manner similar to linked in.

13.Appendix:

13.1 Source code:

app.py

```
import re

from flask import Flask, render_template, request, redirect, url_for, session

import ibm_db

from flask_mail import Mail, Message


app = Flask(__name__)


app.secret_key = "a"

conn = ibm_db.connect(

    "DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=czt98936;PWD=DICkH4DQDETaI9Hm",

    "",

    "",

)
```

```
@app.route("/")
def home():
    return render_template("index.html")



@app.route("/login", methods=["GET", "POST"])
def login():
    global userid
    msg = ""

    if request.method == "POST":
        username = request.form["username"]
        password = request.form["password"]
        sql = "SELECT * FROM users 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"]
            userid = account["USERNAME"]
            session["username"] = account["USERNAME"]
            msg = "Logged in succesfully"

            return render_template("ViewJob.html", msg=msg)
        else:
            msg = "Incorrect username/password!"
    return render_template("index.html", msg=msg)



@app.route("/register", methods=["GET", "POST"])
def register():
    msg = ""
    if request.method == "POST":
        username = request.form["username"]
        email = request.form["email"]
        password = request.form["password"]
        sql = "SELECT * FROM users WHERE username=?"
        stmt = ibm_db.prepare(conn, sql)
```

```
ibm_db.bind_param(stmt, 1, username)
ibm_db.execute(stmt)
account = ibm_db.fetch_assoc(stmt)
print(account)
if account:
    msg = "Account Already exists"
elif not re.match(r"[^@]+@[^@]+\.[^@]+", email):
    msg = "Invalid email"
elif not re.match(r"[A-Za-z0-9]+", username):
    msg = "name must contain only alpha characters or numbers!"
else:
    insert_sql = "INSERT INTO users VALUES(?,?,?)"
    prep_stmt = ibm_db.prepare(conn, insert_sql)
    ibm_db.bind_param(prep_stmt, 1, username)
    ibm_db.bind_param(prep_stmt, 2, email)
    ibm_db.bind_param(prep_stmt, 3, password)
    ibm_db.execute(prep_stmt)
    msg = "you have successfully registered"
    app.config["SECRET_KEY"] = "top-secret!"
    app.config["MAIL_SERVER"] = "smtp.sendgrid.net"
    app.config["MAIL_PORT"] = 587
    app.config["MAIL_USE_TLS"] = True
```

```
        app.config["MAIL_USERNAME"] = "apikey"
        app.config[
            "MAIL_PASSWORD"
        ] = "SG.XbHqaobAQPCL5ZW_3-jRkA.vuaAYWQBDNuRV-
5MjIVERohpOKt-dKvcQmXNsgjFi74"
        app.config["MAIL_DEFAULT_SENDER"] =
"rubantamilboy@gmail.com"
        mail = Mail(app)
        recipient = request.form["email"]
        msg = Message("Successfully Registered",
recipients=[recipient])
        msg.body = (
            "Congratulations! You have successfully registered with "
            "Skill/Job Recommender!"
        )
        msg.html = (
            "<h1>Successfully Registered</h1>"
            "<p>Congratulations! You have successfully registered with "
            "<b>Skill/Job Recommender</b>!</p>"
        )
        mail.send(msg)
        return render_template("index.html")
    elif request.method == "POST":
        msg = "Please fill out the form!"
    return render_template("index.html", msg=msg)



@app.route("/view")
def view():
    return render_template("ViewJob.html")



@app.route("/apply", methods=["GET", "POST"])
def apply():
    msg = ""
    if request.method == "POST":
        username = request.form["username"]
        email = request.form["email"]
        qualification = request.form["qualification"]
        skills = request.form["skills"]
        jobs = request.form["jobs"]

        insert_sql = "INSERT INTO jobs values(?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, username)

        ibm_db.bind_param(prep_stmt, 2, email)

        ibm_db.bind_param(prep_stmt, 3, qualification)

        ibm_db.bind_param(prep_stmt, 4, skills)

        ibm_db.bind_param(prep_stmt, 5, jobs)

        ibm_db.execute(prep_stmt)

        msg = "You have succesfully applied for the job!"

        session["loggedin"] = True

        app.config["SECRET_KEY"] = "top-secret!"

        app.config["MAIL_SERVER"] = "smtp.sendgrid.net"

        app.config["MAIL_PORT"] = 587

        app.config["MAIL_USE_TLS"] = True

        app.config["MAIL_USERNAME"] = "apikey"

        app.config[

            "MAIL_PASSWORD"

        ] = "SG.XbHqaobAQPCL5ZW_3-jRkA.vuaAYWQBDNuRV-
5MjIVERohpOKt-dKvcQmXNsgjFi74"

        app.config["MAIL_DEFAULT_SENDER"] =
"rubantamilboy@gmail.com"

        mail = Mail(app)

        recipient = request.form["email"]

        msg = Message("Successfully Applied", recipients=[recipient])
        msg.body = (
            "Congratulations! You have successfully applied your job with "
            "Skill/Job Recommender!"
        )
        msg.html = (
            "<h1>Successfully Applied</h1>"
            "<p>Congratulations! You have successfully applied your job
with "
            "<b>Skill/Job Recommender</b>!</p>"
        )
        mail.send(msg)
        return redirect(url_for("login"))

    elif request.method == "POST":
        msg = "Please fill the form!"
    return render_template("ApplyJob.html", msg=msg)



@app.route("/logout")
def logout():
    session.pop("loggedin", None)
    session.pop("id", None)
    session.pop("username", None)
    return render_template("index.html")
```

index.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>Registration and Login</title>
        <script defer
src="https://use.fontawesome.com/releases/v5.15.4/js/all.js"
                integrity="sha384-
rOA1PnstxnOBLzCLMcre8ybwbTmemjzdNlILg8O7z1lUkLXozs4DHo
nlDtnE7fpc"
                crossorigin="anonymous"></script>
        <style>
                @import
url('https://fonts.googleapis.com/css?family=Montserrat:400,800');


                * {
```

```css
	box-sizing: border-box;
}

body {
	background: #f6f5f7;
	display: flex;
	justify-content: center;
	align-items: center;
	flex-direction: column;
	font-family: 'Montserrat', sans-serif;
	height: 100vh;
	margin: -20px 0 50px;
}

h1 {
	font-weight: bold;
	margin: 0;
}

h2 {
	text-align: center;
```

```
}

p {
	font-size: 14px;
	font-weight: 100;
	line-height: 20px;
	letter-spacing: 0.5px;
	margin: 20px 0 30px;
}

span {
	font-size: 12px;
}

a {
	color: #333;
	font-size: 14px;
	text-decoration: none;
	margin: 15px 0;
}
```

```
button {
	border-radius: 20px;
	border: 1px solid #41d3ff;
	background-color: #41d3ff;
	color: #FFFFFF;
	font-size: 12px;
	font-weight: bold;
	padding: 12px 45px;
	letter-spacing: 1px;
	text-transform: uppercase;
	transition: transform 80ms ease-in;
}

button:active {
	transform: scale(0.95);
}

button:focus {
	outline: none;
}
```

```
button.ghost {
        background-color: transparent;
        border-color: #FFFFFF;
}

form {
        background-color: #FFFFFF;
        display: flex;
        align-items: center;
        justify-content: center;
        flex-direction: column;
        padding: 0 50px;
        height: 100%;
        text-align: center;
}

input {
        background-color: #eee;
        border: none;
        padding: 12px 15px;
        margin: 8px 0;
```

```
	width: 100%;
}

.container {
	background-color: #fff;
	border-radius: 10px;
	box-shadow: 0 14px 28px rgba(0, 0, 0, 0.25),
			0 10px 10px rgba(0, 0, 0, 0.22);
	position: relative;
	overflow: hidden;
	width: 768px;
	max-width: 100%;
	min-height: 480px;
}

.form-container {
	position: absolute;
	top: 0;
	height: 100%;
	transition: all 0.6s ease-in-out;
}
```

```
.sign-in-container {
        left: 0;
        width: 50%;
        z-index: 2;
}

.container.right-panel-active .sign-in-container {
        transform: translateX(100%);
}

.sign-up-container {
        left: 0;
        width: 50%;
        opacity: 0;
        z-index: 1;
}

.container.right-panel-active .sign-up-container {
        transform: translateX(100%);
        opacity: 1;
        z-index: 5;
        animation: show 0.6s;
}

@keyframes show {

        0%,
        49.99% {
                opacity: 0;
                z-index: 1;
        }

        50%,
        100% {
                opacity: 1;
                z-index: 5;
        }
}

.overlay-container {
        position: absolute;
```

```
            top: 0;
            left: 50%;
            width: 50%;
            height: 100%;
            overflow: hidden;
            transition: transform 0.6s ease-in-out;
            z-index: 100;
        }

        .container.right-panel-active .overlay-container {
            transform: translateX(-100%);
        }

        .overlay {
            background: #41d3ff;
            background: -webkit-linear-gradient(to right, #41d3ff,
#FF416C);
            background: linear-gradient(to right, #41d3ff,
#FF416C);
            background-repeat: no-repeat;
            background-size: cover;
            background-position: 0 0;
        color: #FFFFFF;
        position: relative;
        left: -100%;
        height: 100%;
        width: 200%;
        transform: translateX(0);
        transition: transform 0.6s ease-in-out;
}

.container.right-panel-active .overlay {
        transform: translateX(50%);
}

.overlay-panel {
        position: absolute;
        display: flex;
        align-items: center;
        justify-content: center;
        flex-direction: column;
        padding: 0 40px;
        text-align: center;
```

```
	top: 0;
	height: 100%;
	width: 50%;
	transform: translateX(0);
	transition: transform 0.6s ease-in-out;
}

.overlay-left {
	transform: translateX(-20%);
}

.container.right-panel-active .overlay-left {
	transform: translateX(0);
}

.overlay-right {
	right: 0;
	transform: translateX(0);
}

.container.right-panel-active .overlay-right {
```

```
                transform: translateX(20%);
            }

            .social-container {
                margin: 20px 0;
            }

            .social-container a {
                border: 1px solid #DDDDDD;
                border-radius: 50%;
                display: inline-flex;
                justify-content: center;
                align-items: center;
                margin: 0 5px;
                height: 40px;
                width: 40px;
            }
        </style>
</head>

<body>
```

```
        <h2>Welcome To Skill/Job Recommender Application</h2>
        <div class="container" id="container">
            <div class="form-container sign-up-container">
                <form action="/register" method="post">
                    <h1>Create Account</h1>
                    <span>or use your email for registration</span>
                    <input type="text" placeholder="Name"
name="username" />
                    <input type="email" placeholder="Email"
name="email" />
                    <input type="password" placeholder="Password"
name="password" />
                    <button>Sign Up</button>
                </form>
                <div class="note mt-3 text-center">
                    <!--Register form -->
                    <p> Already have an account ? Please Login <a
href="/login">login! </a> </p>
                </div>
            </div>
            <div class="form-container sign-in-container">
                <form action="/login" method="post">
                    <div class="msg">{{ msg }}</div>
```

```
                        <h1>Sign in</h1>
                        <span>or use your account</span>
                        <input type="text" placeholder="Name"
name="username" />
                        <input type="text" placeholder="Password"
name="password" />
                        <button><a style="color: #fff;">Sign
In</button></a>
                    </form>
                </div>
                <div class="overlay-container">
                    <div class="overlay">
                        <div class="overlay-panel overlay-left">
                            <h1>Welcome Back!</h1>
                            <p>To keep connected with us please login
with your personal info</p>
                            <button class="ghost" id="signIn">Sign
In</button>
                        </div>
                        <div class="overlay-panel overlay-right">
                            <h1>Hello, Friend!</h1>
                            <p>Enter your personal details and start
journey with us</p>
                            <button class="ghost" id="signUp">Sign
```

```
Up</button>
                        </div>
                    </div>
                </div>
            </div>

            <script>
                const signUpButton = document.getElementById('signUp');
                const signInButton = document.getElementById('signIn');
                const container = document.getElementById('container');

                signUpButton.addEventListener('click', () => {
                    container.classList.add("right-panel-active");
                });

                signInButton.addEventListener('click', () => {
                    container.classList.remove("right-panel-active");
                });
            </script>
</body>
```

```
</html>
```

ViewJob.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>View Job</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.mi
n.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwN
LD69Npy4HI+N" crossorigin="anonymous" />

    <style>
      @import url(https://fonts.googleapis.com/css?family=Hind);
      @import url(https://fonts.googleapis.com/css?family=Open+Sans);

      body {
        font-family: 'Open Sans', sans-serif;
        color: #31383d;
        font-size: 1.1rem;
      }

      section {
        padding: 30px 0;
        background: #fff;
      }

      p {
        line-height: 1.5;
        margin: 30px 0;
      }

      h1,
      h2,
      h3,
      h4,
      h5,
      h6 {
        font-family: 'Hind', sans-serif;
        font-weight: 800;
```

```
}

a {
  color: #31383d;
  -webkit-transition: all 0.2s;
  -moz-transition: all 0.2s;
  transition: all 0.2s;
}

a:hover,
a:focus {
  color: #73a5fc;
}

#navbar-main {
  position: absolute;
  font-family: 'Hind', sans-serif;
  background-color: #fff;
  border-bottom: 1px solid #d7e2e9;
}
```

```
#navbar-main .navbar-brand {
  color: #96a4b1;
}


#navbar-main .navbar-toggler {
  padding: 0.5rem;
  border: none;
}


#navbar-main .navbar-toggler-icon {
  background-image: url("data:image/svg+xml;charset=utf8,%3Csvg
viewBox='0 0 30 30' xmlns='http://www.w3.org/2000/svg'%3E%3Cpath
stroke='rgb(68, 189, 255)' stroke-width='2' stroke-linecap='round'
stroke-miterlimit='10' d='M4 8h24M4 16h24M4
24h24'/%3E%3C/svg%3E");
}


#navbar-main .navbar-nav > li.nav-item > a {
  text-transform: uppercase;
  font-size: 0.875rem;
  font-weight: 600;
  letter-spacing: 1px;
  text-align: center;
}

#navbar-main .navbar-brand .fa-cube {
  font-size: 2rem;
}

header.masthead {
  padding-top: 4rem;
  padding-bottom: 4rem;
  margin-bottom: 3rem;
  background: #a3bded;
  background: -webkit-linear-gradient(-20deg, #a3bded 0%, #6991c7
100%);
  background: -moz-linear-gradient(-20deg, #a3bded 0%, #6991c7
100%);
  background: linear-gradient(-20deg, #a3bded 0%, #6991c7 100%);
}

header.masthead .site-heading {
  padding: 100px 0 50px;
  color: #fff;
```

```
}

header.masthead .site-heading h1 {
  font-size: 2.3rem;
}

header.masthead .site-heading .subheading {
  display: block;
  font-weight: 300;
  margin: 0.625rem 0 0;
  color: #fff;
}

ul.job-list {
  margin: 0;
  padding: 0;
  list-style: none;
}

ul.job-list > li.job-preview {
  background: #fff;
  border: 1px solid #d7e2e9;
  -webkit-border-radius: 0.4rem;
  -moz-border-radius: 0.4rem;
  border-radius: 0.4rem;
  padding: 1.5rem 2rem;
  margin-bottom: 1rem;
  float: left;
  width: 100%;
  -webkit-transition: all 0.3s ease 0s;
  -moz-transition: all 0.3s ease 0s;
  transition: all 0.3s ease 0s;
}

ul.job-list > li.job-preview:hover {
  cursor: pointer;
  -webkit-box-shadow: 0 3px 8px rgba(0, 0, 0, 0.05);
  -moz-box-shadow: 0 3px 8px rgba(0, 0, 0, 0.05);
  box-shadow: 0 3px 8px rgba(0, 0, 0, 0.05);
}

.job-title {
  margin-top: 0.6rem;
}

.company {
  color: #96a4b1;
}

.job-preview .btn {
  margin-top: 1.1rem;
}

.btn-apply {
  text-transform: uppercase;
  font-size: 0.875rem;
  font-weight: 800;
  letter-spacing: 1px;
  background-color: transparent;
  color: #393a5f;
  border: 2px solid #393a5f;
  padding: 0.6rem 2rem;
  -webkit-border-radius: 2rem;
  -moz-border-radius: 2rem;
  border-radius: 2rem;
}

.btn-apply:hover {
  background-color: #393a5f;
  color: #fff;
  border: 2px solid #393a5f;
}

@media (max-width: 575px) {
  .job-preview .content {
    width: 100%;
  }
}

@media only screen and (min-width: 992px) {
  #navbar-main {
    background: transparent;
    border-bottom: 1px solid transparent;
  }
```

```
#navbar-main .navbar-brand {
  color: #fff;
  opacity: 0.8;
  padding: 0.95rem 1.2rem;
}

#navbar-main .navbar-brand:hover,
#navbar-main .navbar-brand:focus {
  opacity: 1;
}

#navbar-main .navbar-nav > li.nav-item > a {
  color: #fff;
  opacity: 0.8;
  padding: 0.95rem 1.2rem;
}

#navbar-main .navbar-nav > li.nav-item > a:hover,
#navbar-main .navbar-nav > li.nav-item > a:focus {
  opacity: 1;
```

```
        }
      }
    </style>
  </head>

  <body>
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container-fluid">
        <a class="main-logo-img mt-3" href="/registration and
login.html"><img src="https://skilandjobassignment.s3.jp-tok.cloud-
object-storage.appdomain.cloud/undraw_job_hunt_re_q203.svg"
alt="logo" style="border-radius: 10px" width="100px" /> </a>
        <div class="row donate-sponsor">
          <a type="button" class="btn btn-danger mr-1" id="donate"
href="/logout">Logout</a>
        </div>
      </div>
    </nav>
    <header class="masthead">
      <div class="container">
        <div class="row">
          <div class="col-md-10 offset-md-1">
            <div class="site-heading">
            <h1 class="heading">Open Positions</h1>
            <span class="subheading"> Current listings for jobs. </span>
           </div>
          </div>
         </div>
        </div>
       </header>


       <section>
        <div class="container">
         <div class="row">
          <div class="col-md-10 offset-md-1">
           <ul class="job-list">
            <li class="job-preview">
             <div class="content float-left">
              <h4 class="job-title">Senior Web Designer</h4>
              <h5 class="company">Seattle, WA</h5>
             </div>
             <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
            </li>
            <li class="job-preview">
        <div class="content float-left">
          <h4 class="job-title">Front-End Engineer</h4>
          <h5 class="company">New York, NY</h5>
        </div>
        <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
      </li>
      <li class="job-preview">
        <div class="content float-left">
          <h4 class="job-title">UI/UX Designer</h4>
          <h5 class="company">Los Angeles, CA</h5>
        </div>
        <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
      </li>
      <li class="job-preview">
        <div class="content float-left">
          <h4 class="job-title">Software Developer</h4>
          <h5 class="company">New York, NY</h5>
        </div>
        <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
      </li>
```

```
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Web Developer</h4>
                  <h5 class="company">Los Angeles, CA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Web Designer &amp; Developer</h4>
                  <h5 class="company">Seattle, WA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Junior Programmer</h4>
                  <h5 class="company">Seattle, WA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
```

```
              </li>
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Visual Designer</h4>
                  <h5 class="company">Los Angeles, CA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
              <li class="job-preview">
                <div class="content float-left">
                  <h4 class="job-title">Senior Programmer</h4>
                  <h5 class="company">Seattle, WA</h5>
                </div>
                <a href="/apply" class="btn btn-apply float-sm-right float-xs-
left"> Apply </a>
              </li>
            </ul>
          </div>
        </div>
      </div>
    </section>
    <script>
      window.watsonAssistantChatOptions = {
        integrationID: '7c500335-c4a6-48a7-820b-80449bc10e0e', // The
ID of this integration.
        region: 'au-syd', // The region your integration is hosted in.
        serviceInstanceID: 'c096c74a-c1fc-422b-b91b-c1f68b210cb5', //
The ID of your service instance.
        onLoad: function (instance) {
          instance.render()
        },
      }
      setTimeout(function () {
        const t = document.createElement('script')
        t.src = 'https://web-
chat.global.assistant.watson.appdomain.cloud/versions/' +
(window.watsonAssistantChatOptions.clientVersion || 'latest') +
'/WatsonAssistantChatEntry.js'
        document.head.appendChild(t)
      })
    </script>
  </body>
</html>
```

Applyjob.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>Apply For Job</title>
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.mi
n.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwN
LD69Npy4HI+N" crossorigin="anonymous" />
  </head>

  <body style="padding: 2rem 0; background-color: #eff5f5">
    <nav class="navbar navbar-expand-lg navbar-light">
      <div class="container-fluid">
        <a class="main-logo-img mt-3" href="/registration and
login.html"><img src="https://skilandjobassignment.s3.jp-tok.cloud-
object-storage.appdomain.cloud/undraw_job_hunt_re_q203.svg"
alt="logo" style="border-radius: 10px; margin-top: 10px"
width="100px" /> </a>
```

```
        <div class="row donate-sponsor">
          <a type="button" class="btn btn-primary mr-1" id="sponsor"
href="/view">View Jobs</a>
          <a type="button" class="btn btn-danger mr-1" id="donate"
href="/logout">Logout</a>
        </div>
      </div>
    </nav>
    <div class="container">
      <div class="msg" style="text-align: center">{{ msg }}</div>
      <h1 style="text-align: center; margin-top: 10px">Apply For
Job</h1>
      <br />
      <div class="row mx-0 justify-content-center">
        <div class="col-md-7 col-lg-5 px-lg-2 col-xl-4 px-xl-0 px-xxl-3"
style="border: 6px solid black; border-radius: 12px">
          <form method="POST" class="w-100 rounded-1 p-4 border bg-
white" action="/apply" enctype="multipart/form-data">
            <label class="d-block mb-4">
              <span class="form-label d-block">Your name</span>
              <input required name="username" type="text" class="form-
control" placeholder="Enter Your Name" />
            </label>
            <label class="d-block mb-4">
              <span class="form-label d-block">Email address</span>
              <input required name="email" type="email" class="form-
control" placeholder="Enter your mail id" />
            </label>
            <label class="d-block mb-4">
              <span class="form-label d-block">Qualification</span>
              <input type="text" name="qualification" class="form-control"
placeholder="Enter your qualification" />
            </label>

            <label class="d-block mb-4">
              <span class="form-label d-block">Skills</span>
              <input type="text" name="skills" class="form-control"
placeholder="Enter your skills" />
            </label>

            <label class="d-block mb-4">
              <span class="form-label d-block">Job Role</span>
              <input type="text" name="jobs" class="form-control"
placeholder="Applying For" />
            </label>
            <div class="mb-3">
              <button type="submit" class="btn btn-primary px-3 rounded-
3">Apply</button>
            </div>
          </form>
        </div>
      </div>
    </div>
  </body>
</html>
```

13.2 Github & project demo link:

https://github.com/IBM-EPBL/IBM-Project-1050-1658337039

https://drive.google.com/drive/folders/1NvK4mXwoSSCXXPlgWoMHTOvXnR-OA2W8