# Skill / Job Recommender Application

**Technology:** Cloud App Development

**Team ID:** PNT2022TMID22224

**Batch No.:** B5-5M1E

# 1. INTRODUCTION

# 1.1 Project Overview

Having lots of skills but wondering which job will best suit you? Don’t need to worry! we have come up with a skill recommender solution through which the fresher or the skilled person can 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.

# 1.2 Purpose

The main aim of the project is to building a model 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.

# 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.** 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 content-based 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.

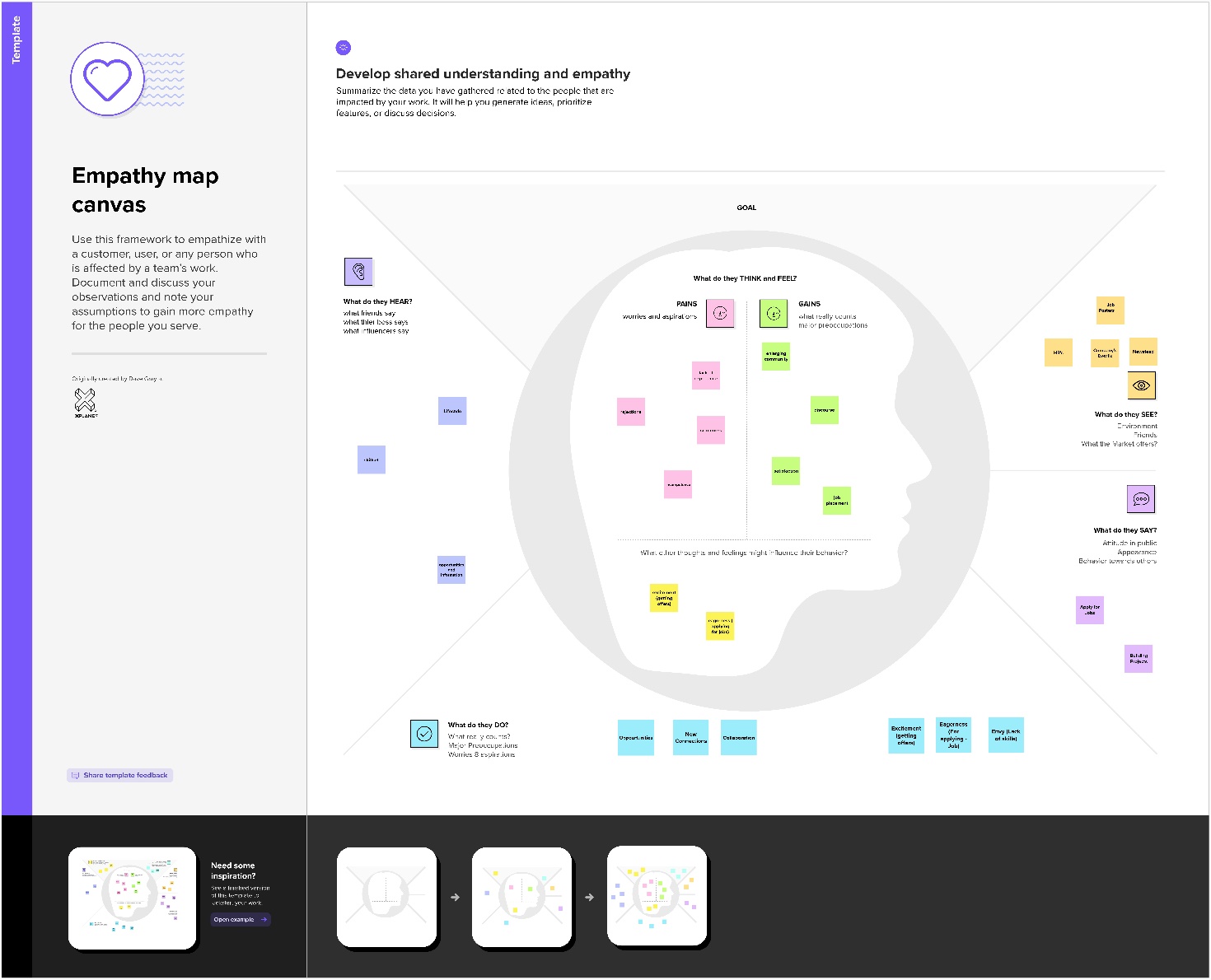
**8.** A. Nigam, A. Roy, H. Singh and H. Waila, "Job Recommendation through Progression of Job Selection," 2019 IEEE 6th International Conference on Cloud Computing and Intelligence Systems (CCIS), 2019, pp. 212-216, doi: 10.1109/CCIS48116.2019.9073723.

# 2.3 Problem Statement Definition

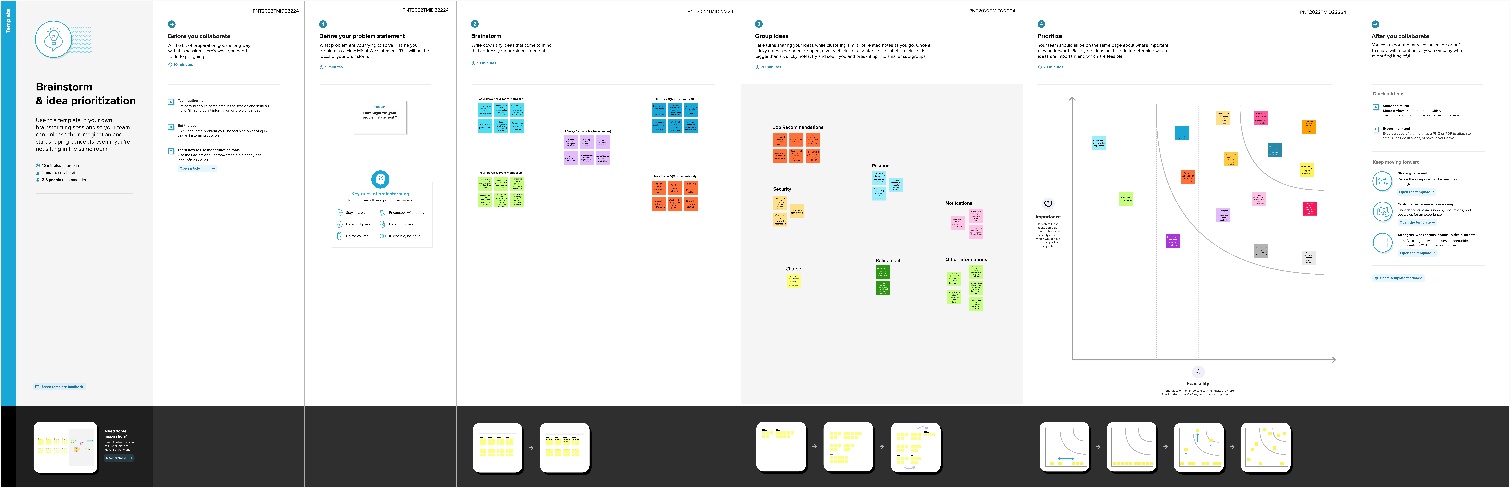
The main aim of the project is to develop an end-to-end web application capable of displaying the current job openings based on the user skill set. Users will interact with the chat-bot and can get recommendations based on their skills. This Application uniquely identifies the user’s skills recommend the job according to the user’s interest.

# 3. IDEATION & PROPOSED SOLUTION

# 3.1 Empathy Map Canvas



# 3.2 Ideation & Brainstorming

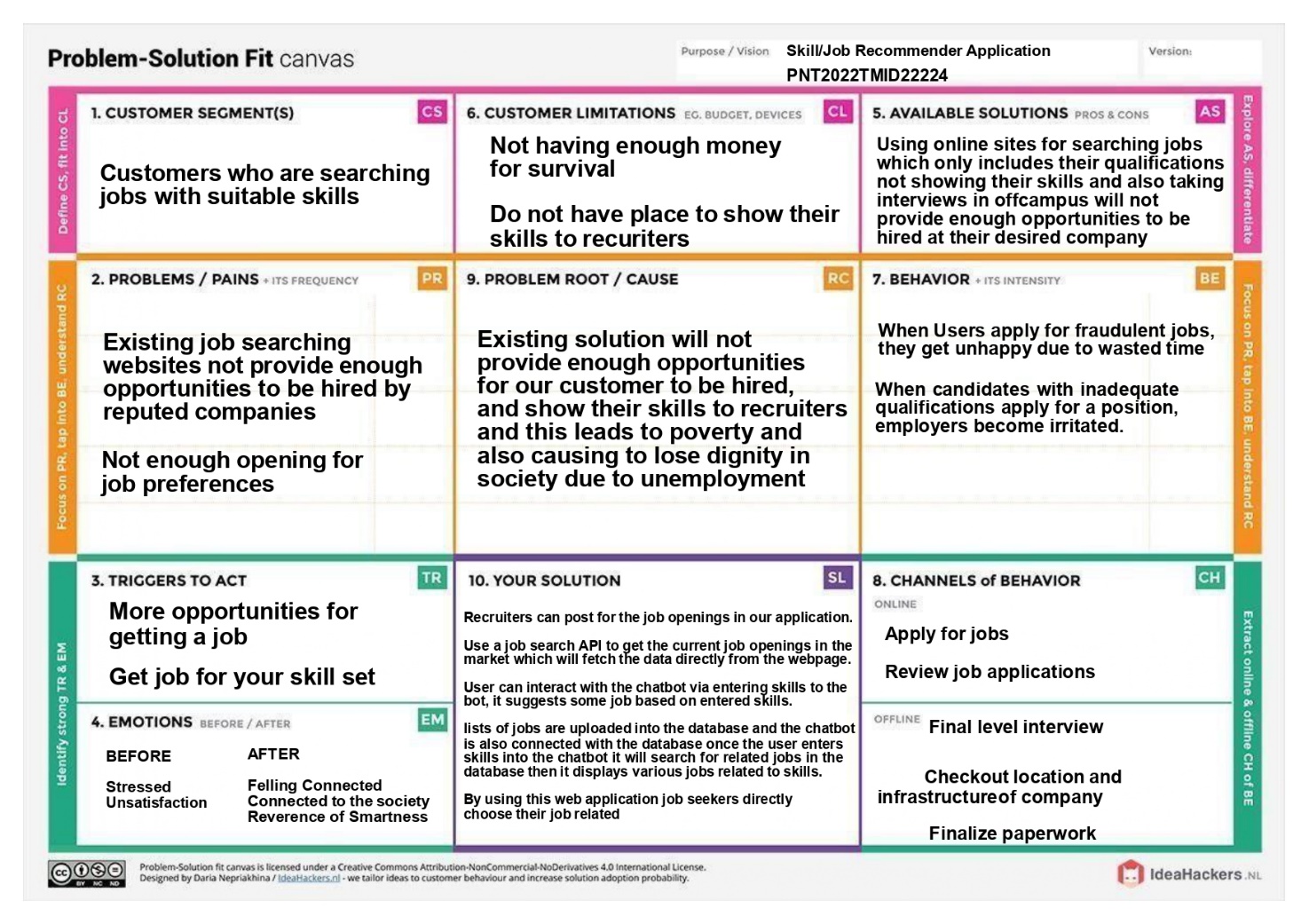


# 3.3 Proposed Solution

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

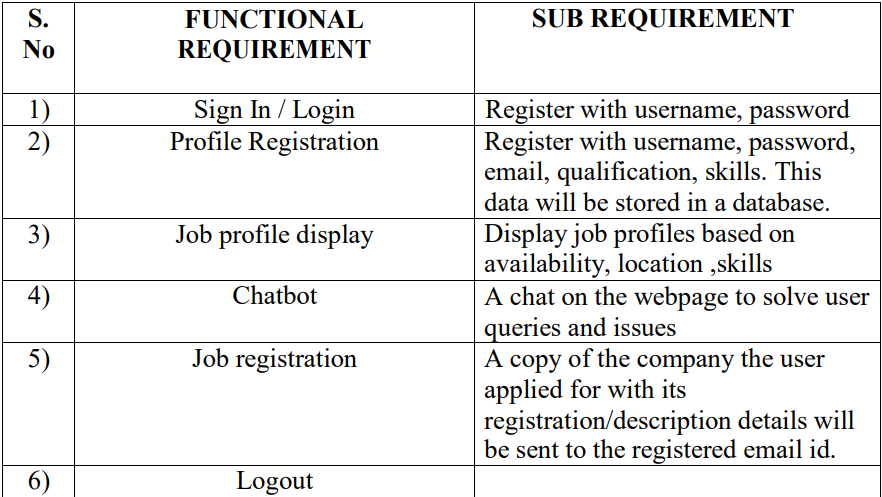
To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage

# 3.4 Problem Solution fit



# 4. REQUIREMENT ANALYSIS

# 4.1 Functional requirement



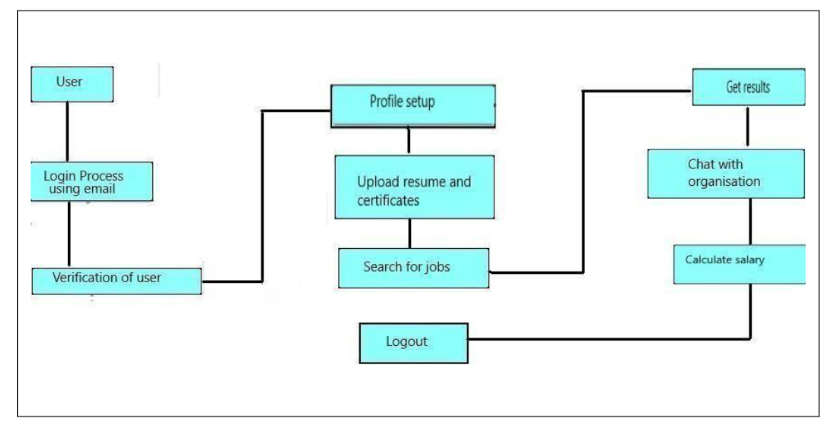
# 4.2 Non-Functional requirements

|  |  |  |
| --- | --- | --- |
| **S. No** | **NON-FUNTIONAL REQUIREMENT** | **DESCRIPTION** |
| 1) | Usability | The webpage will be designed in such a way that any non-technical user can easily navigate through it and complete  the job registration work. |
| 2) | Security | Using of SSL certificate (Python Flask to Cloud connect) will provide security to the project. Database will be safely stored in DB2. |
| 3) | Reliability | To make sure the webpage doesn’t go down due  to network traffic. |
| 4) | Performance | Focus on loading the webpage as  quickly as possible irrespective of the number of user/integrator traffic. |
| 5) | Availability | This webpage will be available to all  users at any given point of time. |
| 6) | Scalability | Increasing the storage space of database can increase  the number of users. |

# 5. PROJECT DESIGN

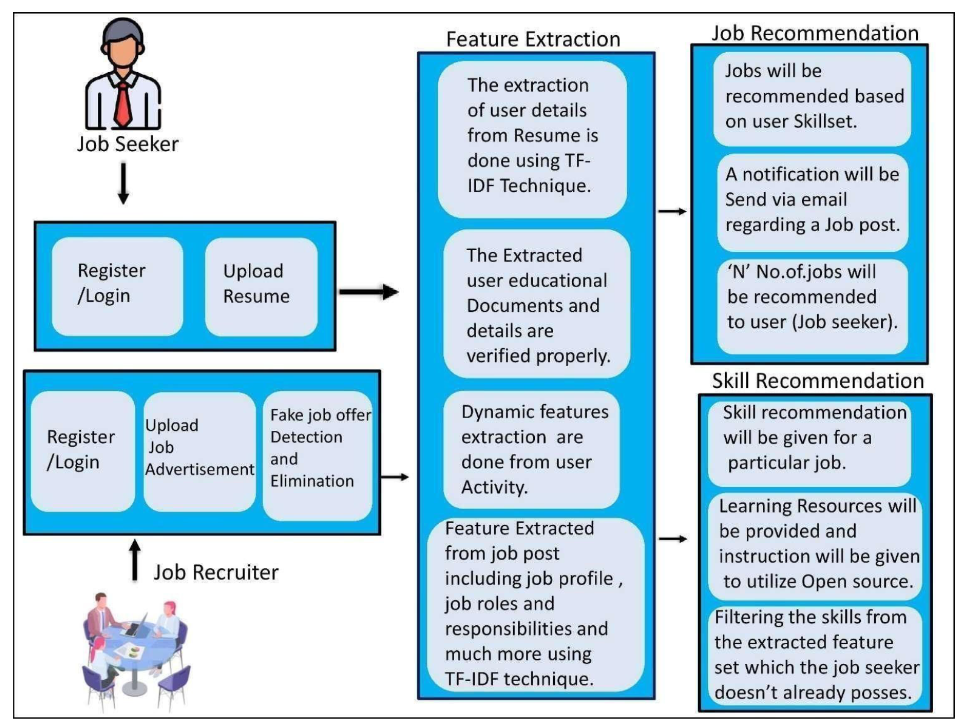
# 5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

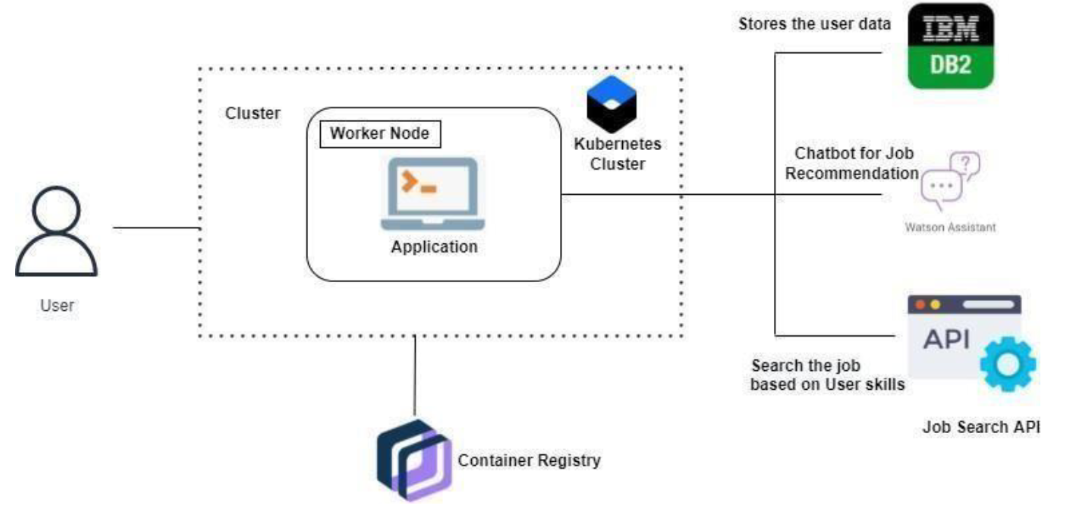


# 5.2 Solution & Technical Architecture

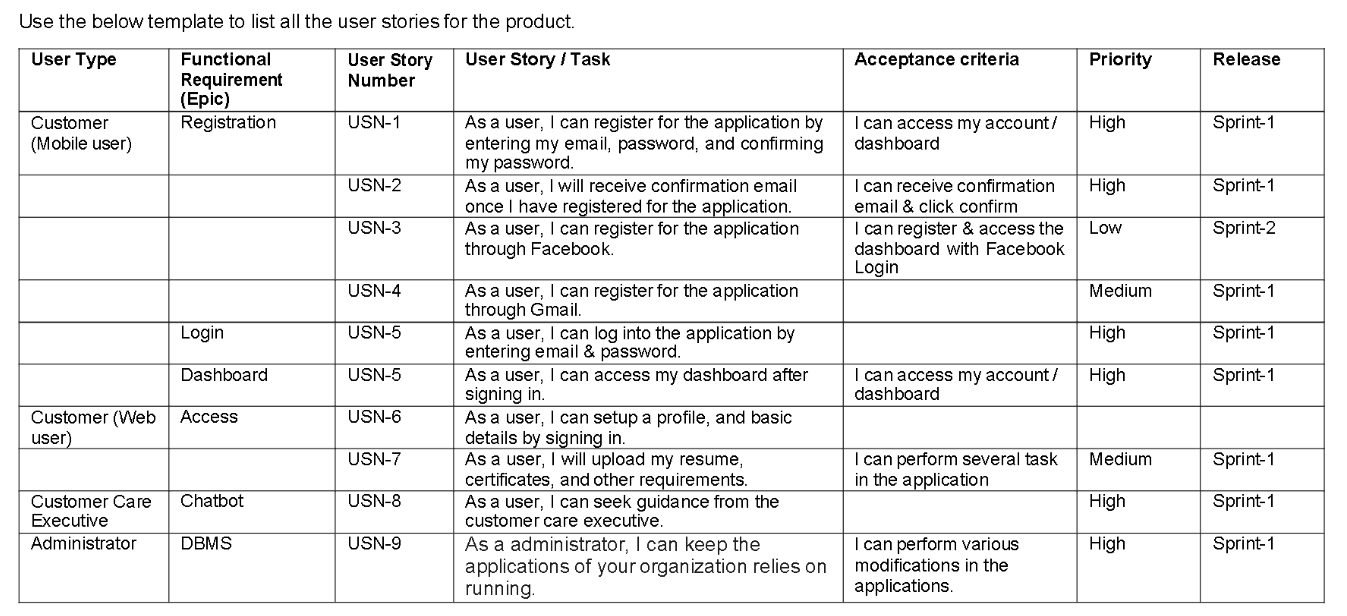
**Solution Architecture:**



**Technical Architecture:**

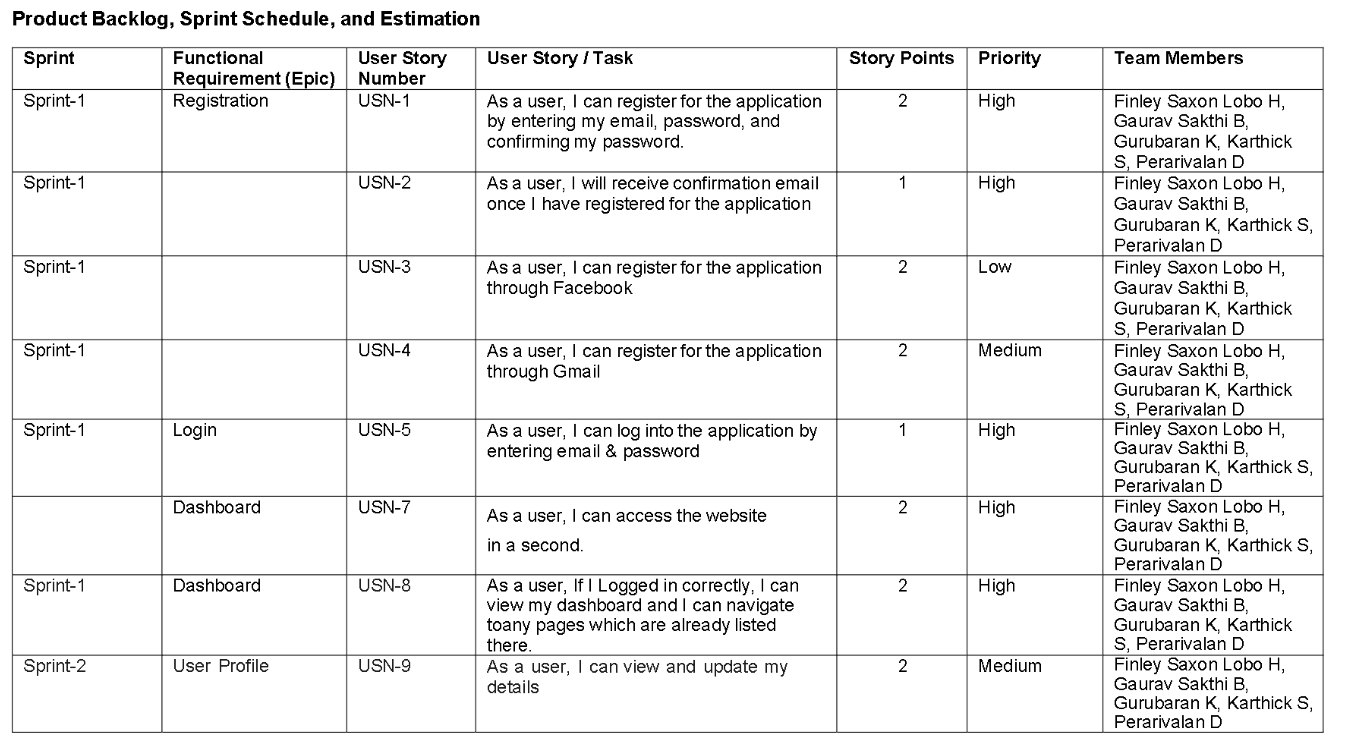


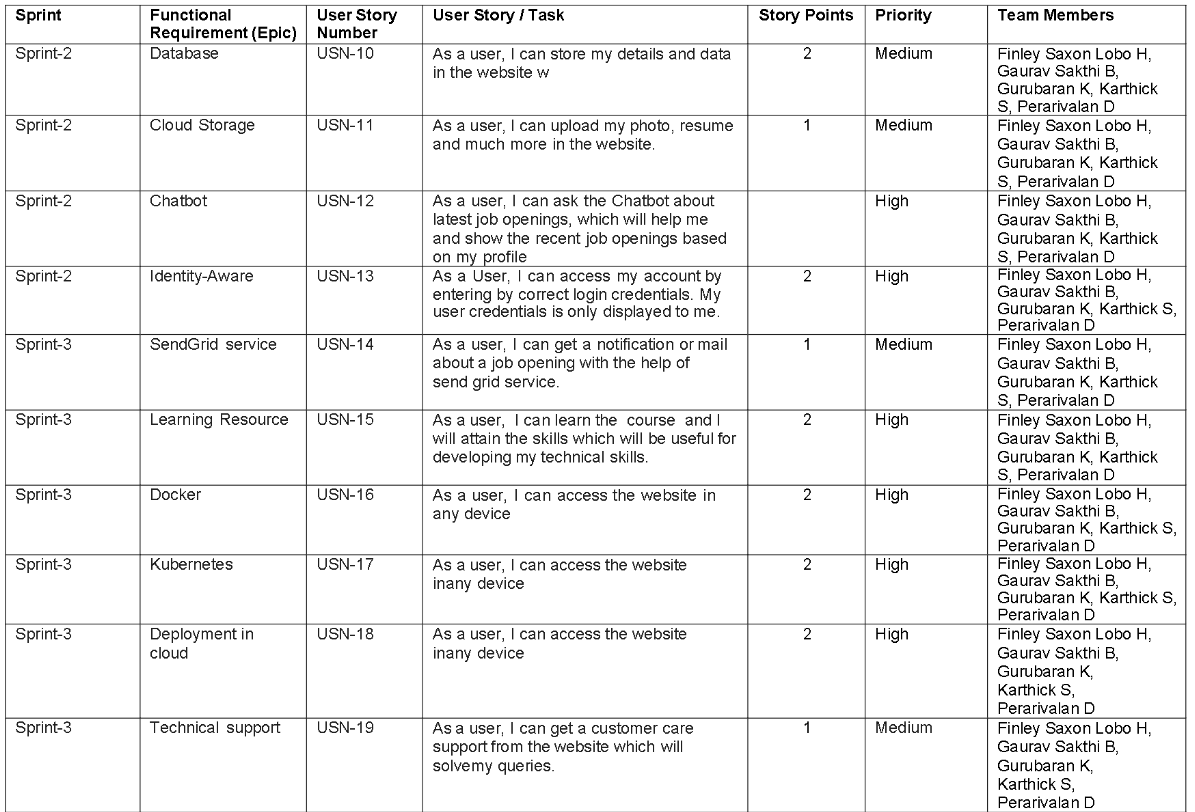
# 5.3 User Stories

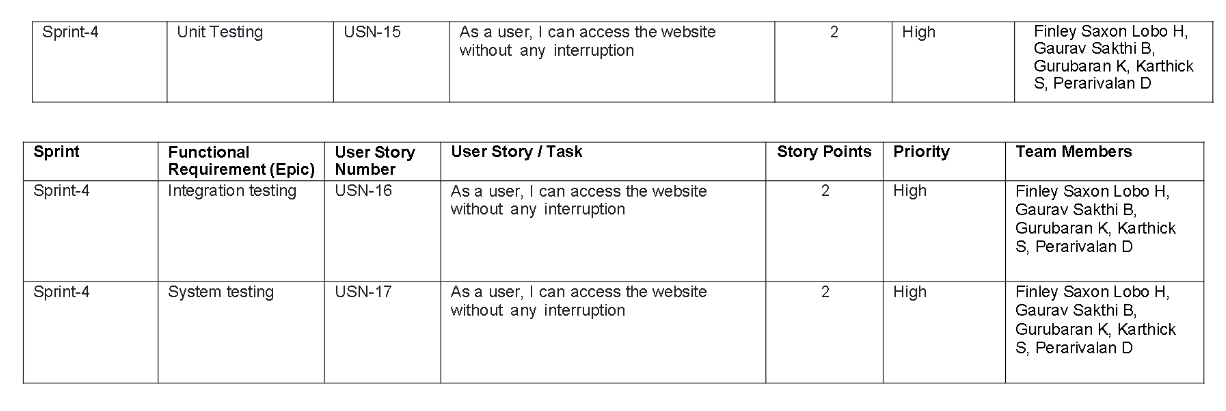


# 6. PROJECT PLANNING & SCHEDULING

# 6.1 Sprint Planning & Estimation







# 6.2 Sprint Delivery Schedule



# 6.3 Reports from JIRA

Average Age Report.

Created vs Resolved Issues Report.

Pie Chart Report.

Recently Created Issues

Report. Resolution Time Report.

Single Level Group By Report.

Time Since Issues Report.

Time Tracking Report.

# 7. CODING & SOLUTIONING

# 7.1 Feature 1

This is one of the feature of our application Skill Pal which provides companies job details for end users

import bcrypt

import ibm\_db

from sendmail import \*

from flask import Flask, redirect, render\_template, request, session, url\_for

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30367;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=vht96126;PWD=YKjRJfgqeiwameyM",'','')

app = Flask(\_\_name\_\_)

app.secret\_key = b'\_5#y2L"F4Q8z\n\xec]/'

@app.route("/",methods=['GET'])

def home():

if 'email' not in session:

return redirect(url\_for('index'))

return render\_template('index.html',name='Home')

@app.route("/index")

def index():

return render\_template('index.html')

@app.route("/index1")

def index1():

return render\_template('index1.html')

@app.route("/job\_details")

def job\_details():

return render\_template('job\_details.html')

@app.route("/job\_details1")

def job\_details1():

return render\_template('job\_details1.html')

@app.route("/job\_details2")

def job\_details2():

return render\_template('job\_details2.html')

@app.route("/job\_details3")

def job\_details3():

return render\_template('job\_details3.html')

@app.route("/job\_details4")

def job\_details4():

return render\_template('job\_details4.html')

@app.route("/job\_details5")

def job\_details5():

return render\_template('job\_details5.html')

@app.route("/job\_details6")

def job\_details6():

return render\_template('job\_details6.html')

@app.route("/job\_listing")

def job\_listing():

return render\_template('job\_listing.html')

@app.route("/about")

def about():

return render\_template('about.html')

@app.route("/registeration",methods=['GET','POST'])

def register():

if request.method == 'POST':

name = request.form['name']

phn = request.form['phn']

email = request.form['email']

psw = request.form['psw']

if not name or not email or not phn or not psw:

return render\_template('registeration.html',error='Please fill all fields')

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

query = "SELECT \* FROM user\_detail WHERE email=? OR phn=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.bind\_param(stmt,2,phn)

ibm\_db.execute(stmt)

print(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO user\_detail(name, email, phn, psw) VALUES (?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, name)

ibm\_db.bind\_param(prep\_stmt, 2, email)

ibm\_db.bind\_param(prep\_stmt, 3, phn)

ibm\_db.bind\_param(prep\_stmt, 4, hash)

ibm\_db.execute(prep\_stmt)

sendMailUsingSendGrid(API,from\_email,to\_emails,subject,html\_content)

return render\_template('registeration.html',success="You can login")

else:

return render\_template('registeration.html',error='Invalid Credentials')

return render\_template('registeration.html',name='Home')

@app.route("/login",methods=['GET','POST'])

def login():

if request.method == 'POST':

email = request.form['email']

psw = request.form['psw']

if not email or not psw:

return render\_template('login.html',error='Please fill all fields')

query = "SELECT \* FROM user\_detail WHERE email=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

print(isUser,psw)

if not isUser:

return render\_template('login.html',error='Invalid Credentials')

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

if not isPasswordMatch:

return render\_template('login.html',error='Invalid Credentials')

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

return redirect(url\_for('home'))

return render\_template('login.html',name='Home')

@app.route("/apply",methods=['GET','POST'])

def apply():

if request.method == 'POST':

name = request.form['name']

email = request.form['email']

psw = request.form['password']

age = request.form['age']

job = request.form['job']

interest = request.form['interest']

if not name or not email or not psw:

return render\_template('apply.html',error='Please fill all fields')

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

query = "SELECT \* FROM applyform WHERE email=? OR psw=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.bind\_param(stmt,2,psw)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO admin\_detail(name, email, psw,age,job,interest) VALUES (?,?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, name)

ibm\_db.bind\_param(prep\_stmt, 2, email)

ibm\_db.bind\_param(prep\_stmt, 3, psw)

ibm\_db.bind\_param(prep\_stmt, 4, age)

ibm\_db.bind\_param(prep\_stmt, 5, job)

ibm\_db.bind\_param(prep\_stmt, 6, interest)

ibm\_db.execute(prep\_stmt)

return render\_template('apply.html',success="You can login")

else:

return render\_template('apply.html',error='Invalid Credentials')

return render\_template('apply.html',name='Home')

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

# 7.2 Feature 2

This chat bot feature provides help tooltip for end users if any help needed for users

<script>

window.watsonAssistantChatOptions = {

integrationID: "65c01ed6-9fc1-4883-979a-3676279ebe44", // The ID of this integration.

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

serviceInstanceID: "8fcd017f-a192-420a-aafc-18cb0330efca", // 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";

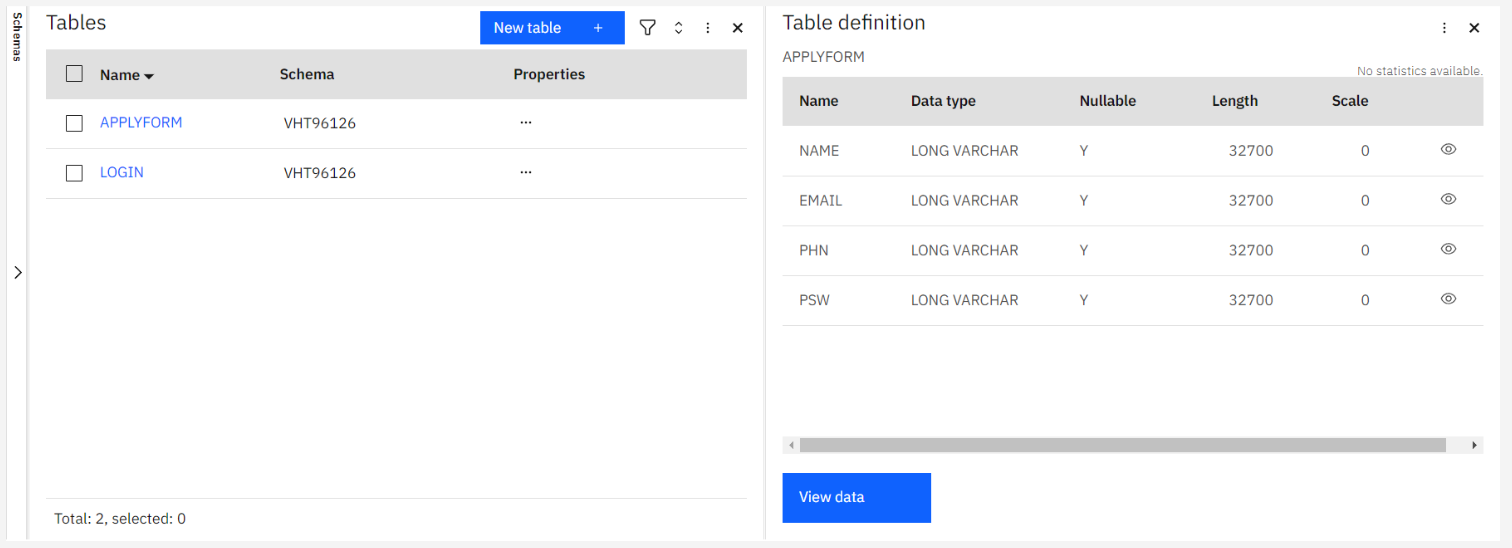
document.head.appendChild(t);

});

</script>

# 7.3 Database Schema (if Applicable)

We user IBM DB2 for our database, below are the tables we used with the parameters given.



# 8. TESTING

# 8.1 Test Cases

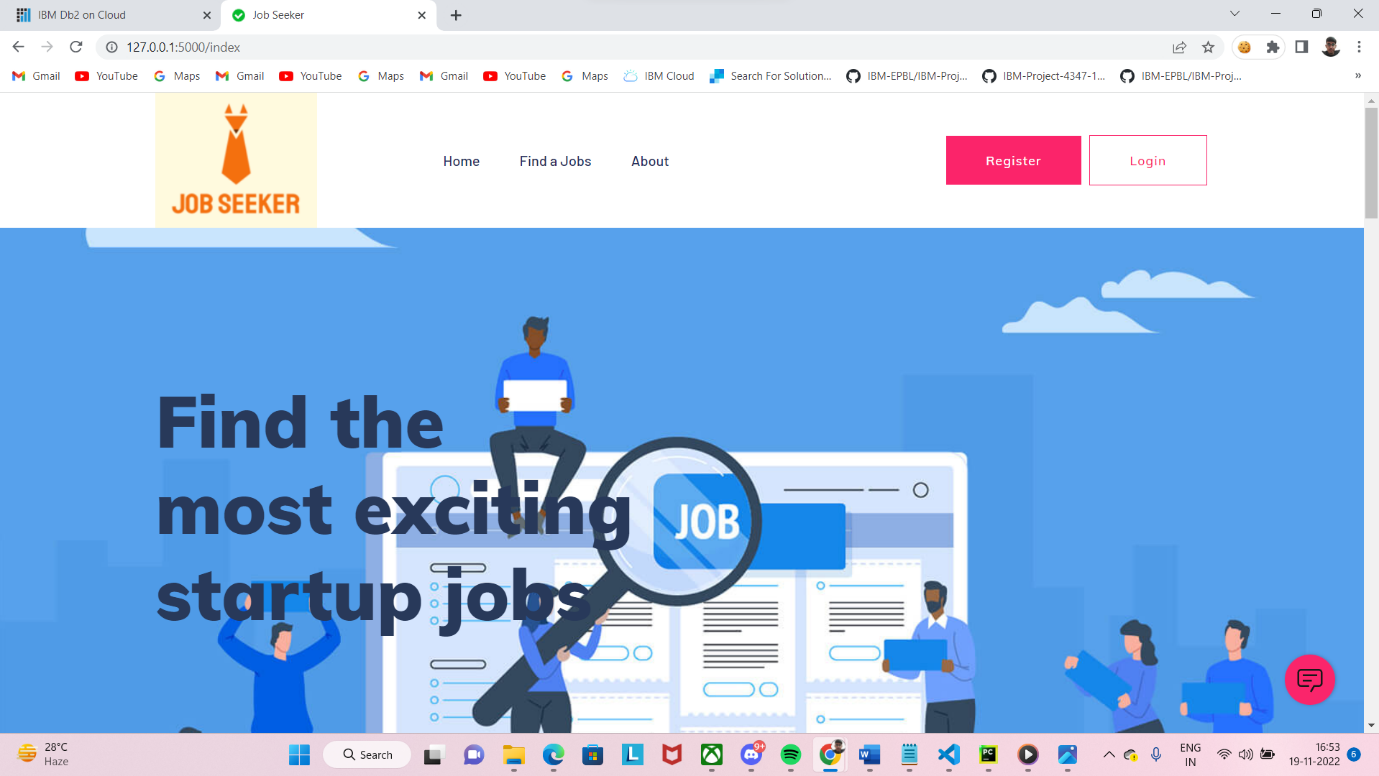
We tested for various validations. Tested all the features with using all the functionalities. Tested the data base storage and retrieval feature too. Testing was done in phase 1 and phase 2, where issues found in phase1 were fixed and then tested again in phase2.

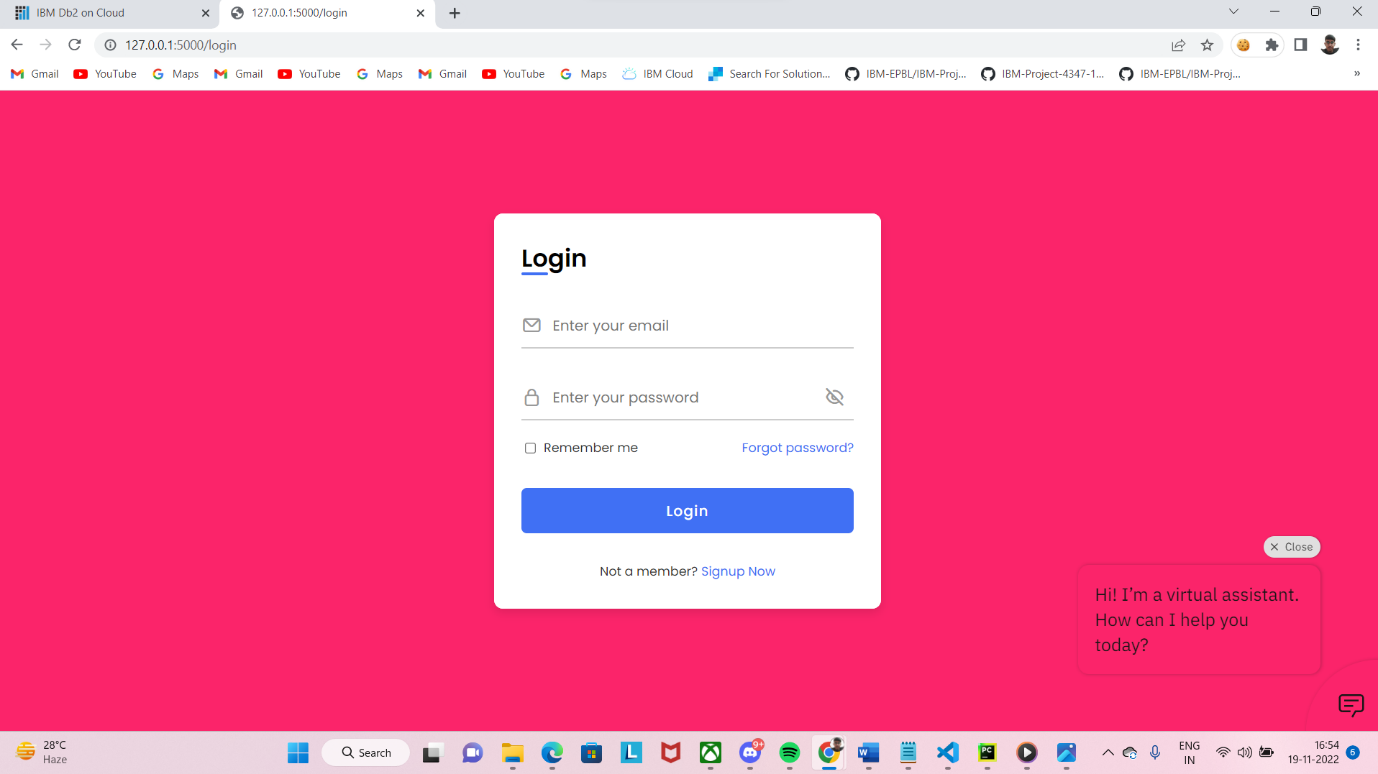
# 8.2 User Acceptance Testing

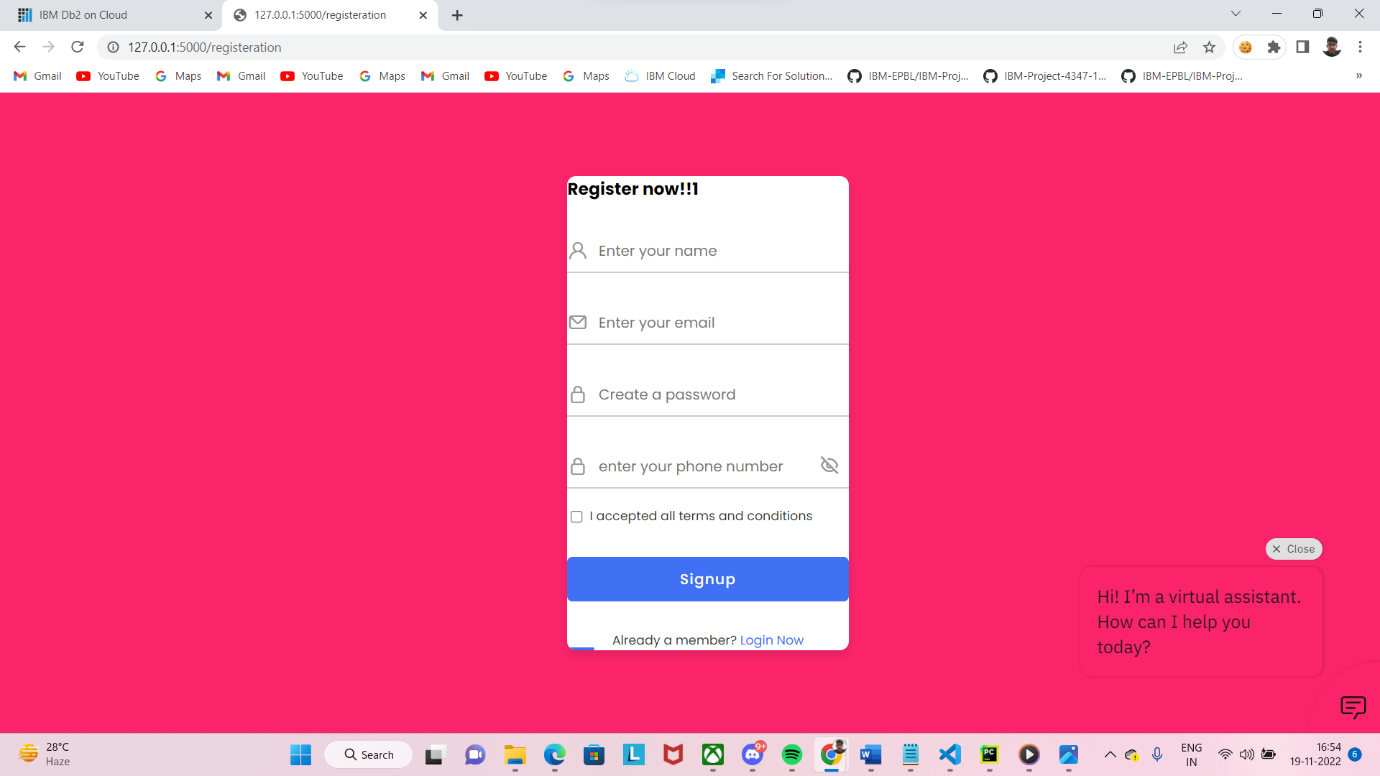
Real world testing was also done, by giving to remote users and asking them to use the application. Their difficulties were fixed and tested again until all the issues were fixed.

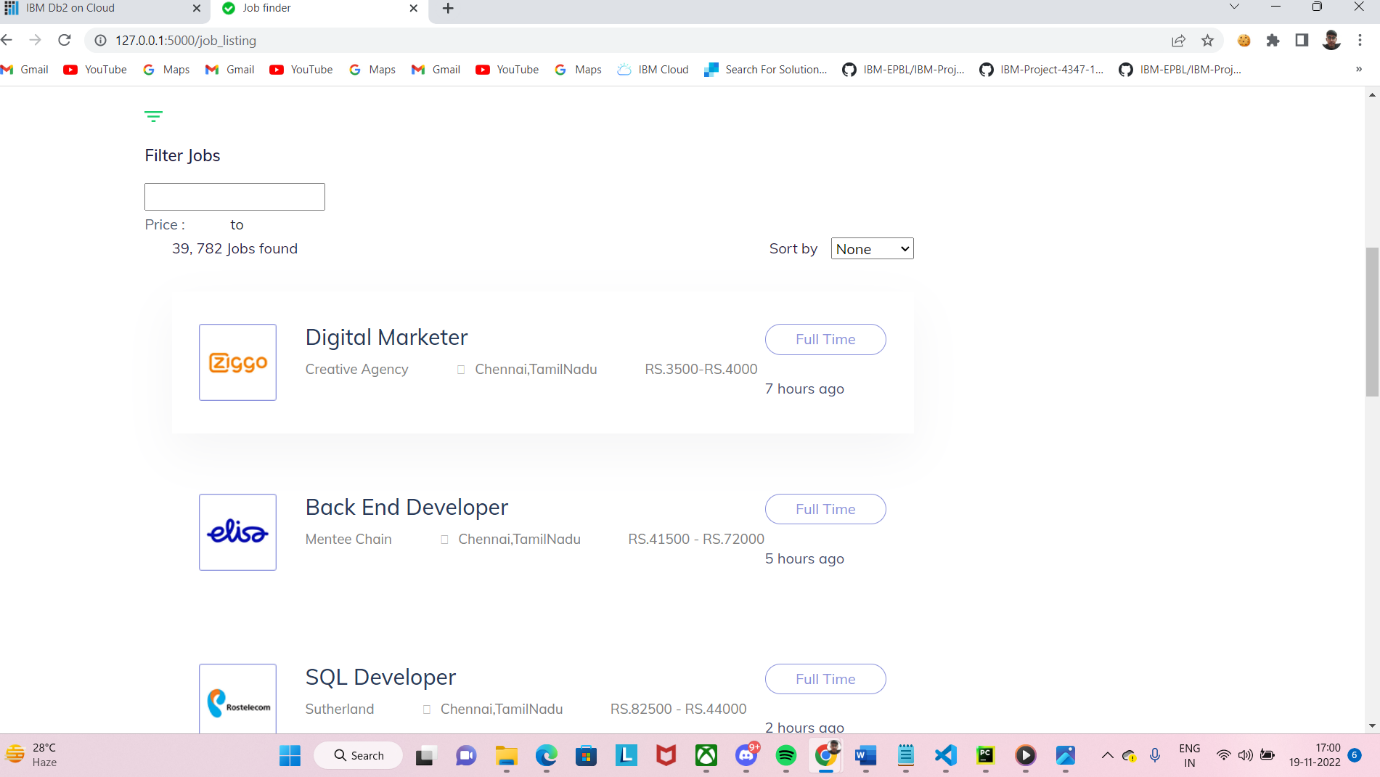
# 9. RESULTS

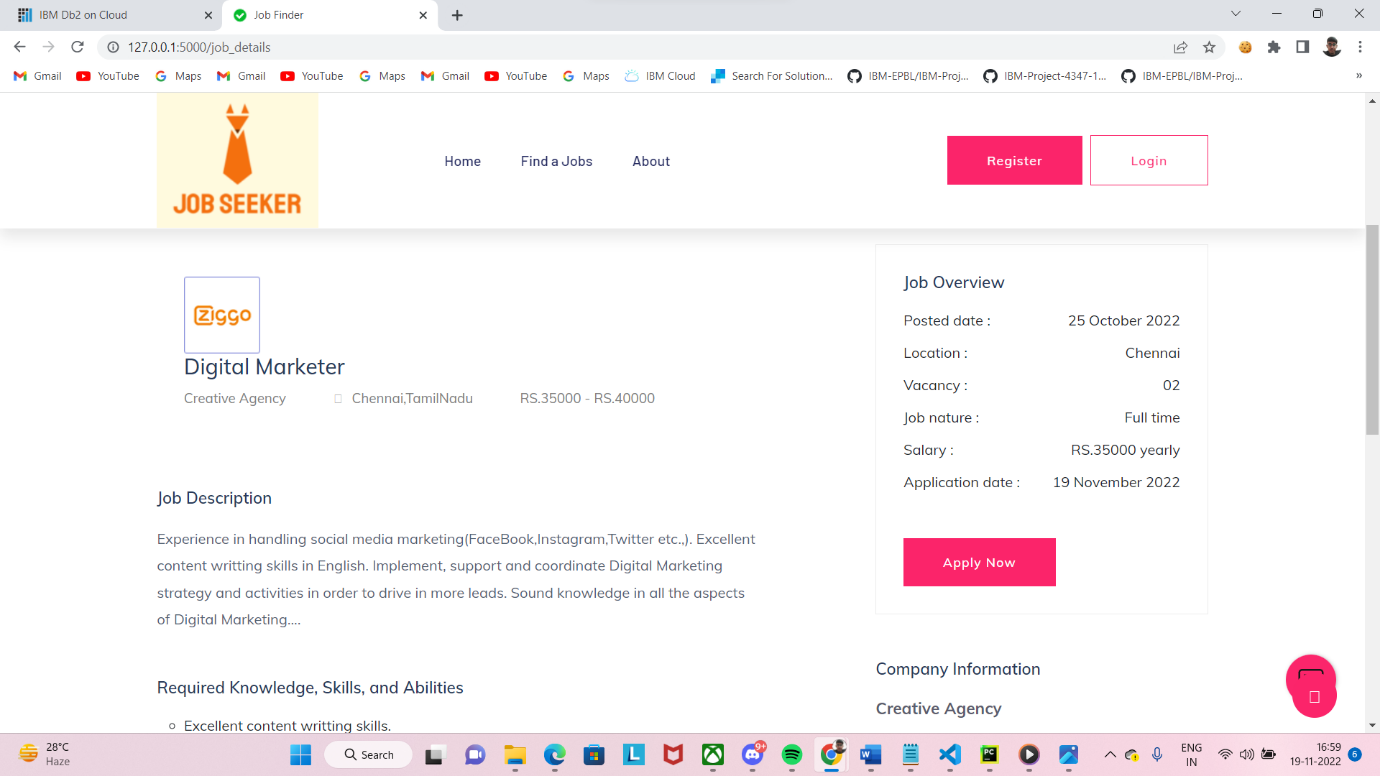
# 9.1 Performance Metrics

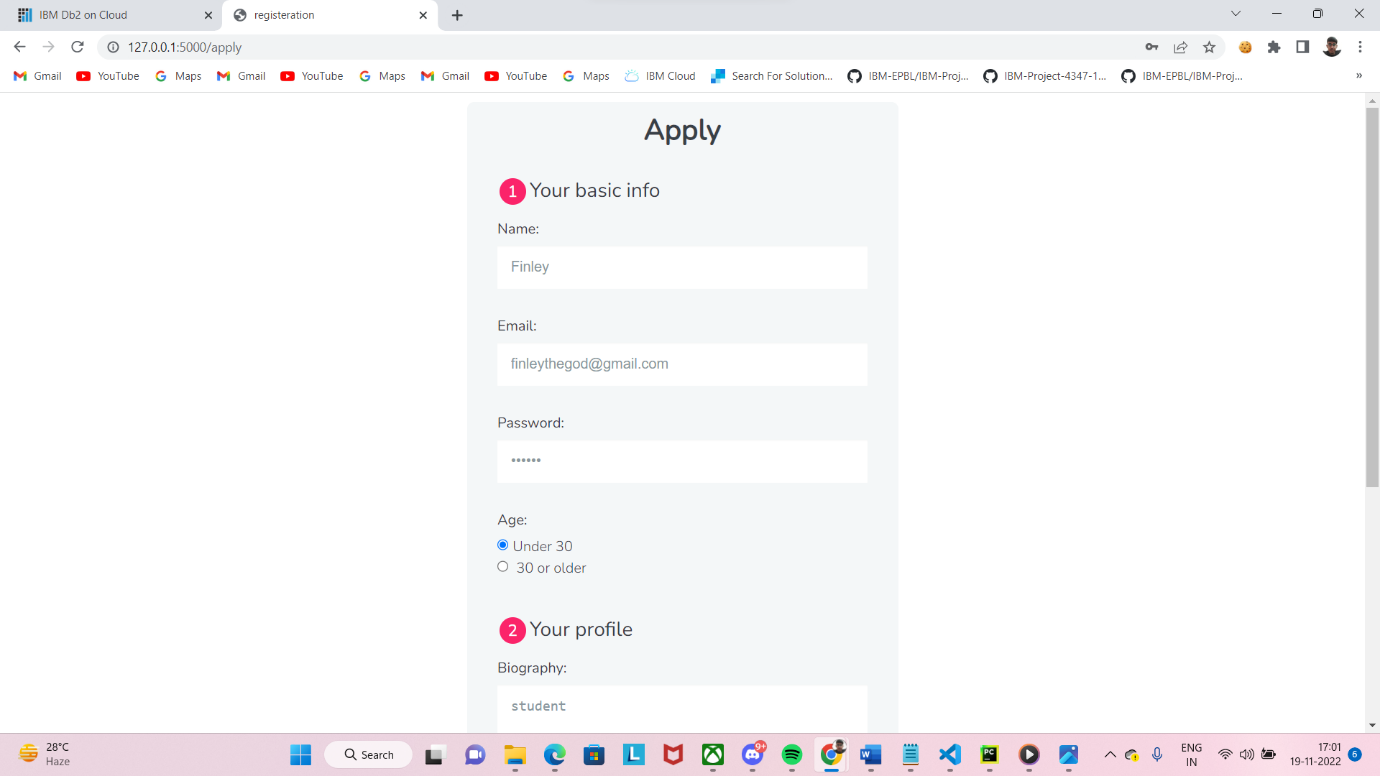


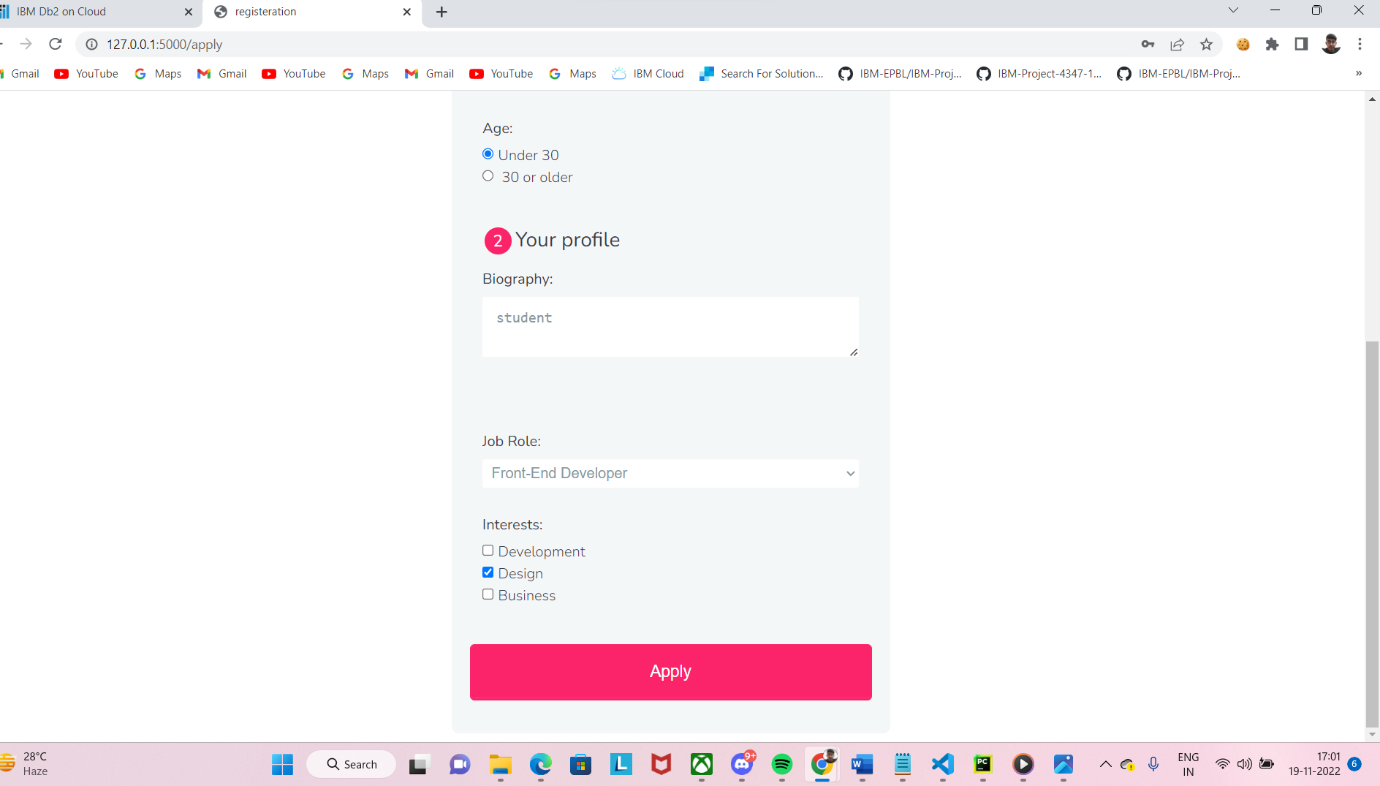












# 10. ADVANTAGES & DISADVANTAGES

**Advantages:**

* + It helps candidates to search the job which perfectly suites them and make them aware of all the job openings.
  + It helps recruiters of the company to choose the right candidates for their organizations with appropriate skills.
  + Since it is cloud application, it does require any installation of software and is portable.

**Disadvantages:**

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

# 11. CONCLUSION

By the end of this project, we will

● know fundamental concepts and techniques of recommender system.

● gain a broad understanding of databases and cloud.

● know how to build a web application using the Flask framework.

● know how to build chatbot.

● know how to containerize the application

# 12. FUTURE SCOPE

The application has been developed to make job search easier. The application that we have developed is user friendly. User can find a job based on their skillset in the short period of time. The jobseeker certainly get benefit by using this application. In the addition, Chatbot Has been implemented with the help of IBM Watson. The chatbot helps jobseeker and organization when they experience the difficulties.

# 13. APPENDIX

# Source Code

**Html code :**

<!doctype html>

<html class="no-js" lang="zxx">

<head>

<meta charset="utf-8">

<meta http-equiv="x-ua-compatible" content="ie=edge">

<title>Job Seeker</title>

<meta name="description" content="">

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

<link rel="manifest" href="site.webmanifest">

<link rel="shortcut icon" type="image/x-icon" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/favicon.ico">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/bootstrap.min.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/owl.carousel.min.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/flaticon.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/price\_rangs.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/slicknav.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/animate.min.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/magnific-popup.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/fontawesome-all.min.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/themify-icons.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/slick.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/nice-select.css">

<link rel="stylesheet" href="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/css/style.css">

<body>

<header>

<!-- Header Start -->

<div class="header-area header-transparrent">

<div class="headder-top header-sticky">

<div class="container">

<div class="row align-items-center">

<div class="col-lg-3 col-md-2">

<!-- Logo -->

<div class="logo">

<a href="index.html"><img src="https://bucketimages.s3.jp-tok.cloud-object-storage.appdomain.cloud/Screenshot%202022-11-19%20164210.png" alt=""></a>

</div>

</div>

<div class="col-lg-9 col-md-9">

<div class="menu-wrapper">

<!-- Main-menu -->

<div class="main-menu">

<nav class="d-none d-lg-block">

<ul id="navigation">

<li><a href="/index">Home</a></li>

<li><a href="/job\_listing">Find a Jobs </a></li>

<li><a href="/about">About</a></li>

</ul>

</nav>

</div>

<!-- Header-btn -->

<div class="header-btn d-none f-right d-lg-block">

<a href="/registeration" class="btn head-btn1">Register</a>

<a href="/login" class="btn head-btn2">Login</a>

</div>

</div>

</div>

<!-- Mobile Menu -->

<div class="col-12">

<div class="mobile\_menu d-block d-lg-none"></div>

</div>

</div>

</div>

</div>

</div>

<!-- Header End -->

</header>

<main>

<!-- slider Area Start-->

<div class="slider-area ">

<!-- Mobile Menu -->

<div class="slider-active">

<div class="single-slider slider-height d-flex align-items-center" data-background="https://bucketimages.s3.jp-tok.cloud-object-storage.appdomain.cloud/background.jpg">

<div class="container">

<div class="row">

<div class="col-xl-6 col-lg-9 col-md-10">

<div class="hero\_\_caption">

<h1>Find the most exciting startup jobs</h1>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- slider Area End-->

<!-- Featured\_job\_start -->

<section class="featured-job-area feature-padding">

<div class="container">

<!-- Section Tittle -->

<div class="row">

<div class="col-lg-12">

<div class="section-tittle text-center">

<span>Recent Job</span>

<h2>Featured Jobs</h2>

</div>

</div>

</div>

<div class="row justify-content-center">

<div class="col-xl-10">

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list1.png" alt="img"></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>Digital Marketer</h4>

</a>

<ul>

<li>Creative Agency</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.35000 - RS.40000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>7 hours ago</span>

</div>

</div>

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list2.png" alt="img"></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>Back End Developer</h4>

</a>

<ul>

<li>Mentee Chain</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.41500 - RS.72000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>5 hours ago</span>

</div>

</div>

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list3.png" alt=""></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>SQL Developer</h4>

</a>

<ul>

<li>Sutherland</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.82500 - RS.44000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>2 hours ago</span>

</div>

</div>

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list4.png" alt=""></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>TIBCO Developer</h4>

</a>

<ul>

<li>Infosys</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.53500 - RS.40000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>1 hours ago</span>

</div>

</div>

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list1.png" alt=""></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>Hadoop Developer</h4>

</a>

<ul>

<li>Capgemini</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.33500 - RS.24000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>3 hours ago</span>

</div>

</div>

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list3.png" alt=""></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>Full stack Developer</h4>

</a>

<ul>

<li>VCW Limited</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.45500 - RS.24000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>4 hours ago</span>

</div>

</div>

<!-- single-job-content -->

<div class="single-job-items mb-30">

<div class="job-items">

<div class="company-img">

<a href="#"><img src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/icon/job-list4.png" alt=""></a>

</div>

<div class="job-tittle job-tittle2">

<a href="#">

<h4>Senior FinOps Engineer</h4>

</a>

<ul>

<li>WPP</li>

<li><i class="fas fa-map-marker-alt"></i>Chennai,TamilNadu</li>

<li>RS.33500 - RS54000</li>

</ul>

</div>

</div>

<div class="items-link items-link2 f-right">

<a href="/job\_details">Full Time</a>

<span>6 hours ago</span>

</div>

</div>

</div>

</section>

<!-- Featured\_job\_end -->

<!-- How Apply Process Start-->

<div class="apply-process-area apply-bg pt-150 pb-150" data-background="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/img/gallery/how-applybg.png">

<div class="container">

<!-- Section Tittle -->

<div class="row">

<div class="col-lg-12">

<div class="section-tittle white-text text-center">

<span>Apply process</span>

<h2> How it works</h2>

</div>

</div>

</div>

<!-- Apply Process Caption -->

<div class="row">

<div class="col-lg-4 col-md-6">

<div class="single-process text-center mb-30">

<div class="process-ion">

<span class="flaticon-search"></span>

</div>

<div class="process-cap">

<h5>1. Search a job</h5>

<p></p>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6">

<div class="single-process text-center mb-30">

<div class="process-ion">

<span class="flaticon-curriculum-vitae"></span>

</div>

<div class="process-cap">

<h5>2. Apply for job</h5>

<p></p>

</div>

</div>

</div>

<div class="col-lg-4 col-md-6">

<div class="single-process text-center mb-30">

<div class="process-ion">

<span class="flaticon-tour"></span>

</div>

<div class="process-cap">

<h5>3. Get your job</h5>

<p></p>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- How Apply Process End-->

</main>

<footer>

<!-- Footer Start-->

<div class="footer-area footer-bg footer-padding">

<div class="container">

<div class="row d-flex justify-content-between">

<div class="col-xl-3 col-lg-3 col-md-4 col-sm-6">

<div class="single-footer-caption mb-50">

<div class="single-footer-caption mb-30">

<div class="footer-tittle">

<h4>About Us</h4>

<div class="footer-pera">

<p> Connect the world’s professionals to make them more productive and successful.</p>

</div>

</div>

</div>

</div>

</div>

<div class="col-xl-3 col-lg-3 col-md-4 col-sm-5">

<div class="single-footer-caption mb-50">

<div class="footer-tittle">

<h4>Contact Info</h4>

<ul>

<li>

<p>Address :<br>

2, jobfinds ltd,

<br>

Anna Nagar,

<br>madurai</p>

</li>

<li><a href="#">Phone : +99999 99999</a></li>

<li><a href="#">Email : info@jobfinder.com</a></li>

</ul>

</div>

</div>

</div>

<!-- footer-bottom area -->

<div class="footer-bottom-area footer-bg">

<div class="container">

<div class="footer-border">

<div class="row d-flex justify-content-between align-items-center">

<div class="col-xl-10 col-lg-10 ">

<div class="footer-copy-right">

<p>

Copyright &copy;<script>document.write(new Date().getFullYear());</script> All rights reserved

</p>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- Footer End-->

</footer>

<!-- JS here -->

<!-- All JS Custom Plugins Link Here here -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/vendor/modernizr-3.5.0.min.js"></script>

<!-- Jquery, Popper, Bootstrap -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/vendor/jquery-1.12.4.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/popper.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/bootstrap.min.js"></script>

<!-- Jquery Mobile Menu -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.slicknav.min.js"></script>

<!-- Jquery Slick , Owl-Carousel Plugins -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/owl.carousel.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/slick.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/price\_rangs.js"></script>

<!-- One Page, Animated-HeadLin -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/wow.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/animated.headline.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.magnific-popup.js"></script>

<!-- Scrollup, nice-select, sticky -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.scrollUp.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.nice-select.min.js"></script>

<script src="./assets/js/jquery.sticky.js"></script>

<!-- contact js -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/contact.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.form.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.validate.min.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/mail-script.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/jquery.ajaxchimp.min.js"></script>

<!-- Jquery Plugins, main Jquery -->

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/plugins.js"></script>

<script src="http://s3.jp-tok.cloud-object-storage.appdomain.cloud/jobportalwebsite/assets/js/main.js"></script>

<script>

window.watsonAssistantChatOptions = {

integrationID: "65c01ed6-9fc1-4883-979a-3676279ebe44", // The ID of this integration.

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

serviceInstanceID: "8fcd017f-a192-420a-aafc-18cb0330efca", // 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>

**Main Code :**

import bcrypt

import ibm\_db

from sendmail import \*

from flask import Flask, redirect, render\_template, request, session, url\_for

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=815fa4db-dc03-4c70-869a-a9cc13f33084.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=30367;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=vht96126;PWD=YKjRJfgqeiwameyM",'','')

app = Flask(\_\_name\_\_)

app.secret\_key = b'\_5#y2L"F4Q8z\n\xec]/'

@app.route("/",methods=['GET'])

def home():

if 'email' not in session:

return redirect(url\_for('index'))

return render\_template('index.html',name='Home')

@app.route("/index")

def index():

return render\_template('index.html')

@app.route("/index1")

def index1():

return render\_template('index1.html')

@app.route("/job\_details")

def job\_details():

return render\_template('job\_details.html')

@app.route("/job\_details1")

def job\_details1():

return render\_template('job\_details1.html')

@app.route("/job\_details2")

def job\_details2():

return render\_template('job\_details2.html')

@app.route("/job\_details3")

def job\_details3():

return render\_template('job\_details3.html')

@app.route("/job\_details4")

def job\_details4():

return render\_template('job\_details4.html')

@app.route("/job\_details5")

def job\_details5():

return render\_template('job\_details5.html')

@app.route("/job\_details6")

def job\_details6():

return render\_template('job\_details6.html')

@app.route("/job\_listing")

def job\_listing():

return render\_template('job\_listing.html')

@app.route("/about")

def about():

return render\_template('about.html')

@app.route("/registeration",methods=['GET','POST'])

def register():

if request.method == 'POST':

name = request.form['name']

phn = request.form['phn']

email = request.form['email']

psw = request.form['psw']

if not name or not email or not phn or not psw:

return render\_template('registeration.html',error='Please fill all fields')

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

query = "SELECT \* FROM user\_detail WHERE email=? OR phn=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.bind\_param(stmt,2,phn)

ibm\_db.execute(stmt)

print(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO user\_detail(name, email, phn, psw) VALUES (?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, name)

ibm\_db.bind\_param(prep\_stmt, 2, email)

ibm\_db.bind\_param(prep\_stmt, 3, phn)

ibm\_db.bind\_param(prep\_stmt, 4, hash)

ibm\_db.execute(prep\_stmt)

sendMailUsingSendGrid(API,from\_email,to\_emails,subject,html\_content)

return render\_template('registeration.html',success="You can login")

else:

return render\_template('registeration.html',error='Invalid Credentials')

return render\_template('registeration.html',name='Home')

@app.route("/login",methods=['GET','POST'])

def login():

if request.method == 'POST':

email = request.form['email']

psw = request.form['psw']

if not email or not psw:

return render\_template('login.html',error='Please fill all fields')

query = "SELECT \* FROM user\_detail WHERE email=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

print(isUser,psw)

if not isUser:

return render\_template('login.html',error='Invalid Credentials')

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

if not isPasswordMatch:

return render\_template('login.html',error='Invalid Credentials')

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

return redirect(url\_for('home'))

return render\_template('login.html',name='Home')

@app.route("/apply",methods=['GET','POST'])

def apply():

if request.method == 'POST':

name = request.form['name']

email = request.form['email']

psw = request.form['password']

age = request.form['age']

job = request.form['job']

interest = request.form['interest']

if not name or not email or not psw:

return render\_template('apply.html',error='Please fill all fields')

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

query = "SELECT \* FROM applyform WHERE email=? OR psw=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.bind\_param(stmt,2,psw)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO admin\_detail(name, email, psw,age,job,interest) VALUES (?,?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, name)

ibm\_db.bind\_param(prep\_stmt, 2, email)

ibm\_db.bind\_param(prep\_stmt, 3, psw)

ibm\_db.bind\_param(prep\_stmt, 4, age)

ibm\_db.bind\_param(prep\_stmt, 5, job)

ibm\_db.bind\_param(prep\_stmt, 6, interest)

ibm\_db.execute(prep\_stmt)

return render\_template('apply.html',success="You can login")

else:

return render\_template('apply.html',error='Invalid Credentials')

return render\_template('apply.html',name='Home')

if \_\_name\_\_ == "\_\_main\_\_":

app.run(debug=True)

# GitHub

**GitHub Link:** <https://github.com/IBM-EPBL/IBM-Project-13756-1659529058>