SKILL/JOB RECOMMENDER APPLICATION

A Project Report Submitted

By

Team Id: PNT2022TMID03974

Pradeep Narayanan – 412519106102

S S V K Sahit-412519106156

Akshai Kumar-412519106301

Naveen Kumar-412519106085

1. **Introduction** :

1.1 Project overview :

Insight of 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.

1.2 **Purpose:**

Before people used to search the jobs in newspaper and advertisements but now using a app we can easily get recruitment in a company .It can solve complex problems in a way that fits the state of your customers. Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior. Sharpen your communication and marketing strategy with the right triggers and messaging. Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems. Understand the existing situation in order to improve it for your target group.

2. **Literature survey:**

2.1Existing Problem:

A recommendation engine, also known as a recommender system, is software that analyzes available data to make suggestions for something that a website user might be interested in, such as a book, a video or a job, among other possibilities. Using a recommender system can aid job seekers in locating positions that are right for them. Job offers are first gathered from job search websites, and then they are processed to remove pertinent information like job titles and technical qualifications. Clusters of job offers are formed based on their shared characteristics. If a job seeker likes one position inside a cluster, he or she is likely to prefer other positions within that cluster as well. After comparing data from job clusters and job seeker activity, which includes user activities like applications, likes, and ratings, a list of the top recommendations is presented and displayed.

2.2References

Jorge Job First March In this paper, Valverde- Recommendation Workshop on 2018 new data Rebaz, based on Job Narrative contains job Ricardo Seeker Skills: An Extraction seeker Puma, Paul Empirical Study From Text profile and Bustios, (Text2Story set of job Nathalia C. 2018) co-

vacancies Silva. located with and 40th recommenda Europeantions are Conference made based on on Information professional Retrieval skills and (ECIR evaluation is 2018)At: performed of Grenoble, two state-of France the-art methods. Amber Job 2019, IEEE Decemb In this paper, Nigam, Recommendation 6th 2019 we introduce Aakash through International a Roy, Progression of Conference methodology Arpan Job Selection on Cloud where we Saxena, Computing leverage the Hartaran and progression Singh Intelligence of job System(CCI selection by S) candidates using machine learning.Recommendations consists of sub-e commendations for both users and job.

Aritra Skill-based Career 2020 IEEE 256 March In this paper, Ghosh, Path Modeling International 2021 we propose a Beverly and Conference novel and Woolf, Recommendation on Big Data interpretable Shlomo (Big Data) monotonic Zilberstein, nonlinear Andrew state-space Lan. model to analyze online user professional profiles and provide actionable feedback and recommenda tions to users on how they can reach their career goals. Bhavya Job International 9 March In this paper, Chawla Recommendation Journal For 2021 we present , Naitik System Research In our Kansara Applied approaches , Sakshie Science and to Pathak Engineering style , Mr. S. B. Technology(I employment Nikam JRASET) recommendation system for a career based social networking websites. We take a bottom-up approach: we start with deeply understanding and exploring the info and gradually build the smaller bits of the system.

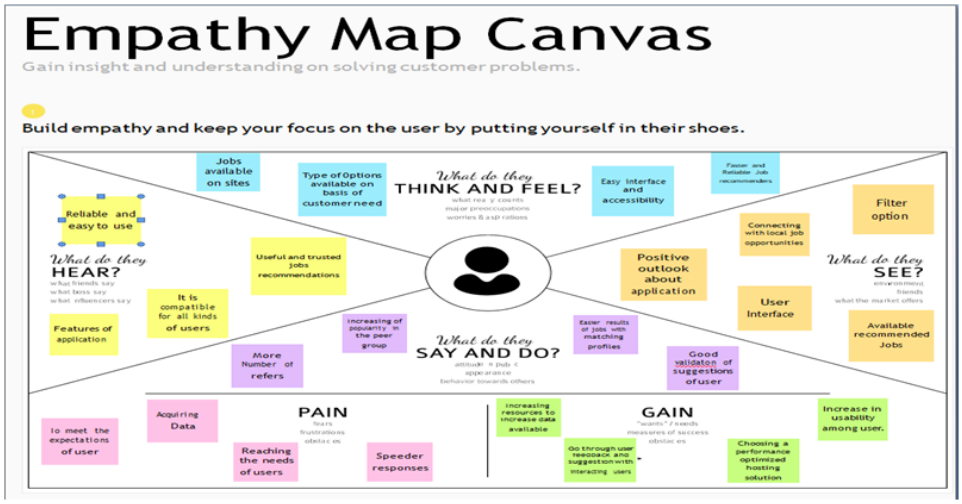
Nikolas Dawson, Mary-Anne Williams, MarianAndrei Rizoiu. Skill-driven recommendations for job transition pathways. Creative Commons Attribution License August 2021 In this paper, They construct a unique Job Transitions Recommend er System that incorporates the skill set distance measures together with other labor market data from job ads and employment statistics. The outputs of system accurately predict transitions between occupations (Accuracy = 76%) and are validated against a dataset of occupational transitions from a longitudinal household survey.

2.3Problem Statement Definition

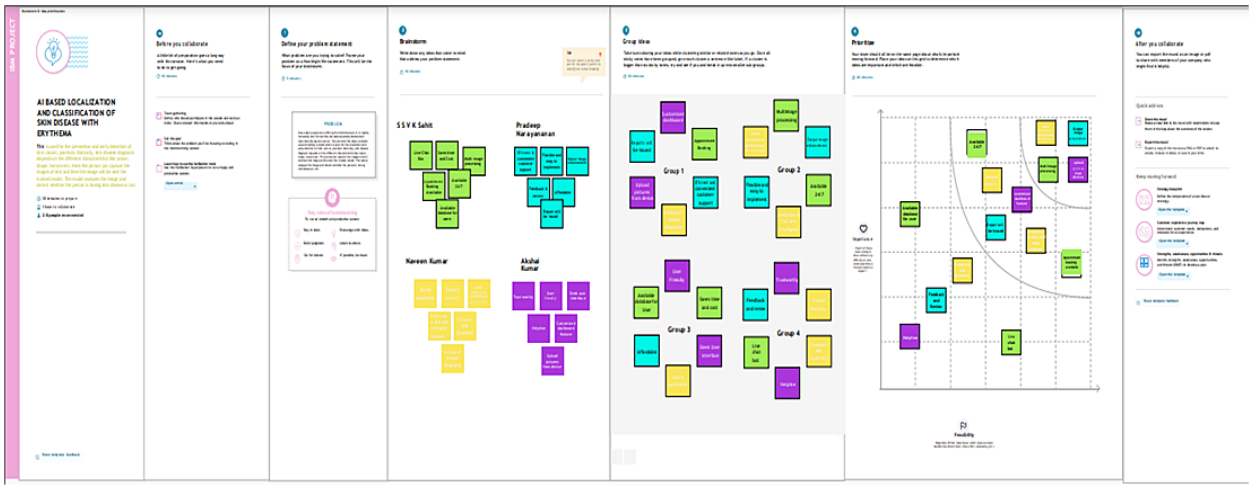
Job seeking is a general issue in the society. So it is difficult for a fresher to look into the job opportunities in newspapers and paper advertisement.So by this they lose hope in themselves.It also happens due to not meeting the eligibility criteria.

3.IDEATION & PROPOSED SOLUTION

3.1Empathy Map Canvas



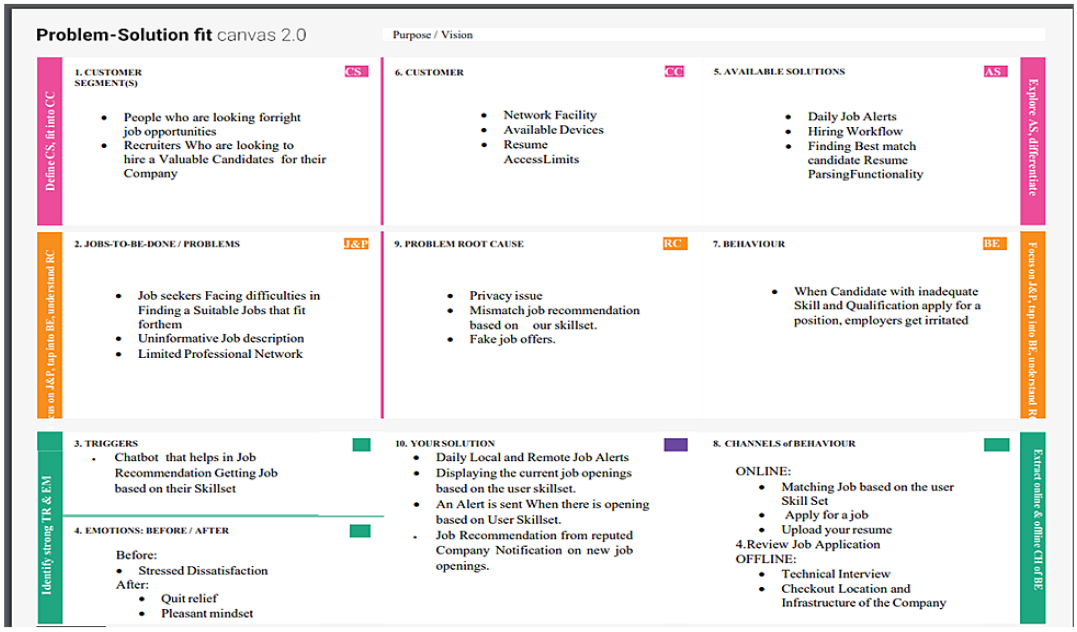
3.2**Ideation & Brainstorming**



3.3**Proposed Solution**

This project proposes to create a web application and deployment using IBM cloud and as we've seen, the modern job seeker must overcome a number of obstacles before locating a position that suits them. Although all of the current work is very promising, there are some faults in other areas. These problems brought up by earlier research must be resolved, and the systems shortcomings must be reduced. A fully functional user interface supporting a job aggregator and recommendation system is planned for development in the proposed system. Every element of the project is made from scratch and in a unique way.

3.4Problem Solution fit



4.REQUIREMENT ANALYSIS

4.1Functional requirement

Following are the functional requirements of the proposed solution

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User confirmation	Confirmation via Email Confirmation via OTP
FR-3	Chat bot	This tool will be there in website which is use to solve the problems related to finding a job , Search for the job and much more
FR-4	User login	By entering your registered data you can login. You can log in using linked Gmail account
FR-5	User search	Searching of jobs is based on job filters and skill recommendation

FR-6	User profile	You can update your profile using Login credentials..
FR-7	User acceptance	Assuring your job by conformation.

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

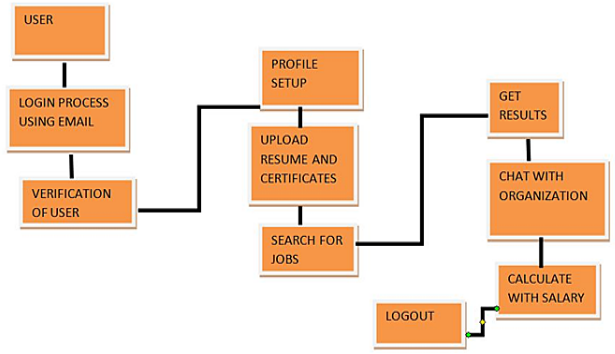
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Usability	This website is used for the job wanted people to login and find job based on their skills. It is also used to fulfil the company's objective to get employees suitable to them
FR-2	Security	This website is safe with one login for job seekers and another login for job Recruiters.
FR-3	Reliability	This website is a User Go one to find jobs and follows the policy of free use no pay website.it recommends job with enormous openings without any limitations.
FR-4	Performance	This website as fast response and take less time to do any process

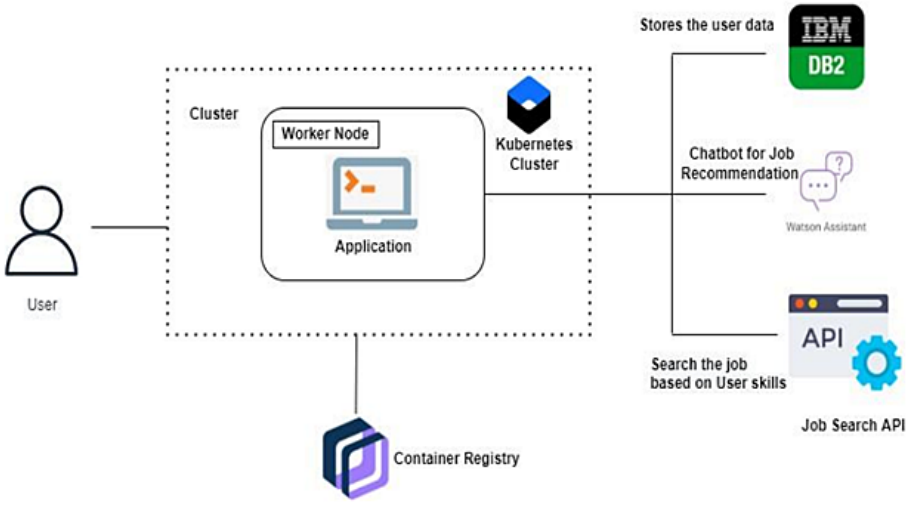
FR-5	Availability	This website provides job offers with suitable skills and it also recommends skills for particular job openings

PROJECT DESIGN

5.1Data 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.





5.3 **User stories**

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register the application by entering my email or password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail.		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password.		High	Sprint-1
	Dashboard	USN-5	As a user,I can access my dashboard after sign up.	I can access my dashboard/account	High	Sprint-1

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Web user)	Access	USN-6	As a user, i can set up profile and basic portfolio by signing in.			
		USN-7	As a user, i will upload my resume and other requirements.	I can perform several tasks	Medium	Sprint-1
Customer Care Executive	Chatbot	USN-8	As a user,i can seek help from the customer care service.		High	Sprint-1
Administrator	DBMS	USN-9	As a user, i can keep the applications of the organization relies on running	I can perform modifications various tasks in application	High	Sprint-1

6.PROJECT PLANNING & SCHEDULING

6.1Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	USER: I can register for the application by entering my email, password, and confirming my password.	10	High	S S V K Sahit Pradeep Narayan Akshai Kumar Naveen Kumar
Sprint-1	Verification	USN-2	USER: I will receive confirmation email for veru	10	High	S S V K Sahit Pradeep Narayan Akshai Kumar Naveen Kumar
Sprint-2	Login	USN-3	USER: I can log into the application by entering email & password	10	Low	S S V K Sahit Pradeep Narayan Akshai Kumar Naveen Kumar
Sprint-2	Verification	USN-4	USER: After click login button, It verify the login credetials wether entered details are correct ornot.	10	Medium	S S V K Sahit Pradeep Narayan Akshai Kumar Naveen Kumar
Sprint-3	Dashboard	USN-5	USER: I can access my dashboard after signing in.	10	High	S S V K Sahit Pradeep Narayan Akshai Kumar NaveenKumar

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-3	Profile	USN-6	USER: I can set up a profile, and basic details.	10	High	S S V K Sahit Pradeep Narayan Akshai Kumar NaveenKumar
Sprint-4	ChatBot	USN-7	USER: I can access the chatbot for job recommendation .	10	High	S S V K Sahit Pradeep Narayan Akshai Kumar NaveenKumar
Sprint-4	Salary	USN-8	USER: I will be able to know the salary formy job	10	High	S S V K Sahit Pradeep Narayan Akshai Kumar NaveenKumar

.

2.Sprint Delivery Schedule

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

BURNDOWN CHART

Days	Goal	Done	Goal velocity	Remaining
0	0	0	500	500
1	50	35	450	465
2	50	22	400	443
3	50	50	350	393
4	50	45	300	348
5	50	49	250	299
6	50	32	200	267
7	50	40	150	227
8	50		100	
9	50		50	
10	50		0	

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

1. User Friendy UI: the UI is designed such a way that each application user is able to understand the working of the application.the application is not complex and simple to use.
2. python flask :it is used for running the application in local server for checking HTML pages are navigated correctly to run a application in a local server.

python code:

```
from flask import Flask, render_template, request, redirect, url_for, session
from flask_mail import Mail,Message
from linkedin_api import Linkedin
import requests
import pandas as pd
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail as mls
import pandas as pd
import csv
import os
import json
import ibm_db
dsn_hostname = "54a2f15b-5c0f-46df-8954-
7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud"
dsn_uid = "mzw99208"
dsn_pwd = "X5OCas8TwAsNxiZ9"
dsn_driver = "{IBM DB2 ODBC DRIVER}"
dsn_database = "BLUDB"
```

```
dsn_port = "32733"
dsn_protocol = "TCPIP"
dsn_security = "SSL"
dsn = (
    "DRIVER={0};"
    "DATABASE={1};"
    "HOSTNAME={2};"
    "PORT={3};"
    "PROTOCOL={4};"
    "UID={5};"
    "PWD={6};"
    "SECURITY={7};").format(dsn_driver, dsn_database, dsn_hostname,
dsn_port, dsn_protocol, dsn_uid, dsn_pwd, dsn_security)
print(dsn)
db=ibm_db
conn = ibm_db.connect(dsn, "", "")
print ("Connected to database: ", dsn_database, "as user: ", dsn_uid, "on
host: ", dsn_hostname)
var_list = []
PROXYCURL_API_KEY = '877e8019097a754907f2eb658b2a44e17374afb4'  # todo - fill
this field up
app = Flask(__name__)
app.secret_key='a'
@app.route('/')
def home():
  return render_template('login.html')
@app.route('/linkpass')
def linkpass():
  return render_template('linked.html')
@app.route('/skillreg',methods=["POST", "GET"])
def dashhome():
  insert_sql = "INSERT INTO SKILLUSER (EMAILID, PASSWORD, NAME,PHONENUMBER)
VALUES (?,?,?,?)"
  prep_stmt = ibm_db.prepare(conn, insert_sql)
  email = request.form['username']
  password= request.form['password']
```

```
    name = request.form['name']
    ph = request.form['phonenum']
    var_list.append(email)
    var_list.append(password)
    var_list.append(name)
    var_list.append(ph)
    ibm_db.bind_param(prep_stmt, 1, email)
    ibm_db.bind_param(prep_stmt, 2, password)
    ibm_db.bind_param(prep_stmt, 3, name)
    ibm_db.bind_param(prep_stmt, 4, ph)
    print("giun")
    ibm_db.execute(prep_stmt)
    return render_template('email.html')

@app.route('/gojob',methods=["POST", "GET"])
def gojob():
    print("job")
    url = "https://indeed11.p.rapidapi.com/"
    p=1
    payload = {
        "search_terms": "Marketing",
        "location": "United States",
        "page": "1"
    }
    headers = {
        "content-type": "application/json",
        "X-RapidAPI-Key":
"9381357d88msha354337c2eb1e98p1348a7jsn192d84997537",
        "X-RapidAPI-Host": "indeed11.p.rapidapi.com"
    }
    response = requests.request("POST", url, json=payload, headers=headers)
    html=response.text
    dict=json.loads(html)
    print(dict)
    return render_template('upgrade.html',data=dict)
@app.route('/register',methods=["POST", "GET"])
```

```
def register():
  return render_template('register.html')
@app.route('/linkedlogin',methods=["POST", "GET"])
def linkedlogin():
  username = request.form['username']
  api_endpoint = 'https://nubela.co/proxycurl/api/v2/linkedin'
  header_dic = {'Authorization': 'Bearer ' + PROXYCURL_API_KEY}
  params = {
      'url': f'https://www.linkedin.com/in/{username}',
  }
  response = requests.get(api_endpoint,
                          params=params,
                          headers=header_dic)
  print(response.json())
  return render_template('skill.html')
@app.route('/confirm',methods=["POST", "GET"])
def confirm():
  print("hi")
  msg = Message('Registration successfully completed', sender =
'pradeepnarayaniyappan@gmail.com', recipients = [var_list[0]])
  msg.body = "Thank You for registering in Skill And Job recommender and
Submit your resume in your profile section and Can apply for your desired
jobs"
  return render_template('Application form.html')
@app.route('/submitapp',methods=["POST", "GET"])
def subapp():
  insert_sql = "INSERT INTO DETAILS
(NAME,FNAME,GENDER,EID,ADDRESS,TENTHMARK,TWELTHMARK,DEG_CGPA,AADHAR,DOMAIN)
VALUES (?,?,?,?,?,?,?,?,?,?)"
  prep_stmt = ibm_db.prepare(conn, insert_sql)
  name = request.form['yourname']
  fname = request.form['fname']
  GD = request.form['GD']
  EID = request.form['EID']
  add = request.form['AL1']
  mark1 = request.form['s5']
```

```
  mark2 = request.form['h5']
  cgpa = request.form['b5']
  AADHAR = request.form['Aadhar']
  DOMAIN = request.form['domain']
  ibm_db.bind_param(prep_stmt, 1,str(
name),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_VARCHAR)
  ibm_db.bind_param(prep_stmt, 2,str
(fname),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_VARCHAR)
  ibm_db.bind_param(prep_stmt, 3,str
(GD),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_VARCHAR)
  ibm_db.bind_param(prep_stmt, 4,str
(EID),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_VARCHAR)
  ibm_db.bind_param(prep_stmt, 5,str
(add),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_VARCHAR)
  ibm_db.bind_param(prep_stmt,6,str
(mark1),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_INTEGER)
  ibm_db.bind_param(prep_stmt,7,str
(mark2),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_INTEGER)
  ibm_db.bind_param(prep_stmt, 8,str
(cgpa),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_INTEGER)
  ibm_db.bind_param(prep_stmt, 9,
str(AADHAR),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_INTEGER)
  ibm_db.bind_param(prep_stmt, 10,str
(DOMAIN),ibm_db.SQL_PARAM_INPUT,ibm_db.SQL_VARCHAR)
 print("giun")
  ibm_db.execute(prep_stmt)
  msg = Message('Your Application has been Saved', sender =
'pradeepnarayaniyappan@gmail.com', recipients = [EID])
  msg.body = "Further you can edit the application form in your profile
section and continue your job searching and apply to it"
  return render_template('skill.html')
@app.route('/skilllogin',methods=["POST", "GET"])
def login():
  msg = ''
  if request.method == 'POST' and 'username' in request.form and 'password'
in request.form:
```

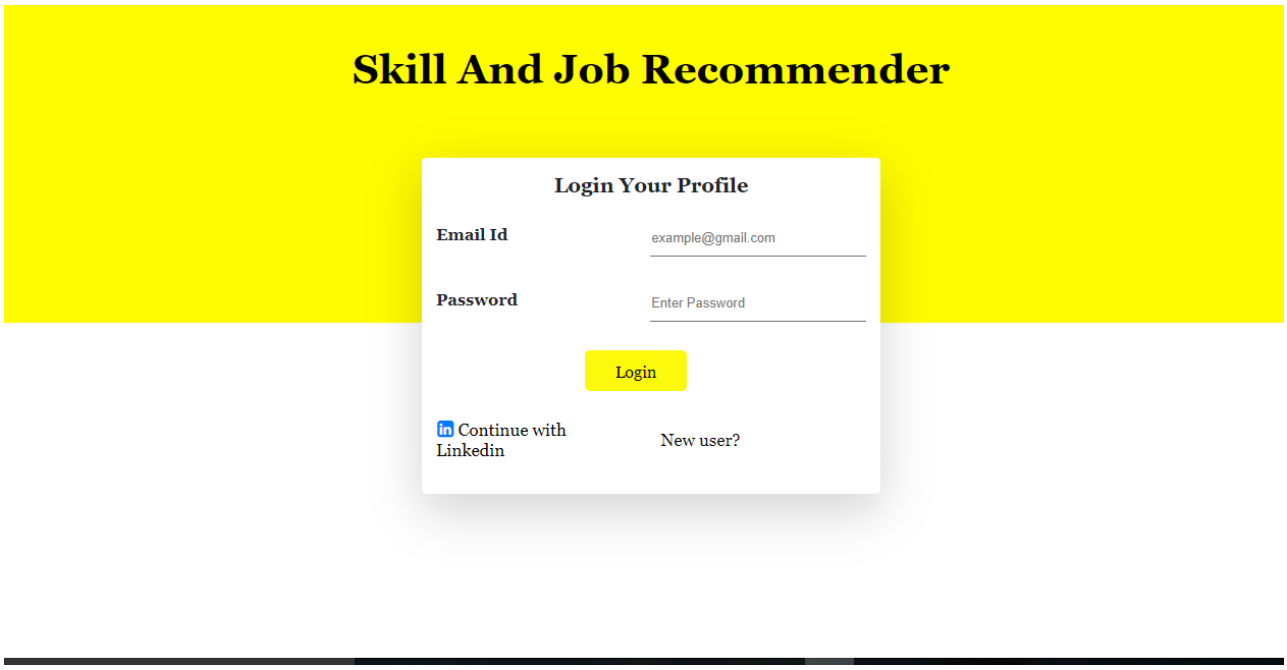
```
        email = request.form['username']
        password = request.form['password']
        sql = "SELECT * FROM SKILLUSER WHERE EMAILID =? AND PASSWORD =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        if account:
            msg = 'Logged in successfully !'
            return render_template('skill.html', msg = msg)
        else:
            msg = 'Incorrect email / password !'
    return render_template('login.html', msg = msg)
if __name__ == '__main__':
    app.run(debug=True)
```

Cloud DB2 : we have used IBM cloud Db2 for user data storage and we have given all API credentials in the above program integrated with html page to store the data in IBM Cloud DB2.

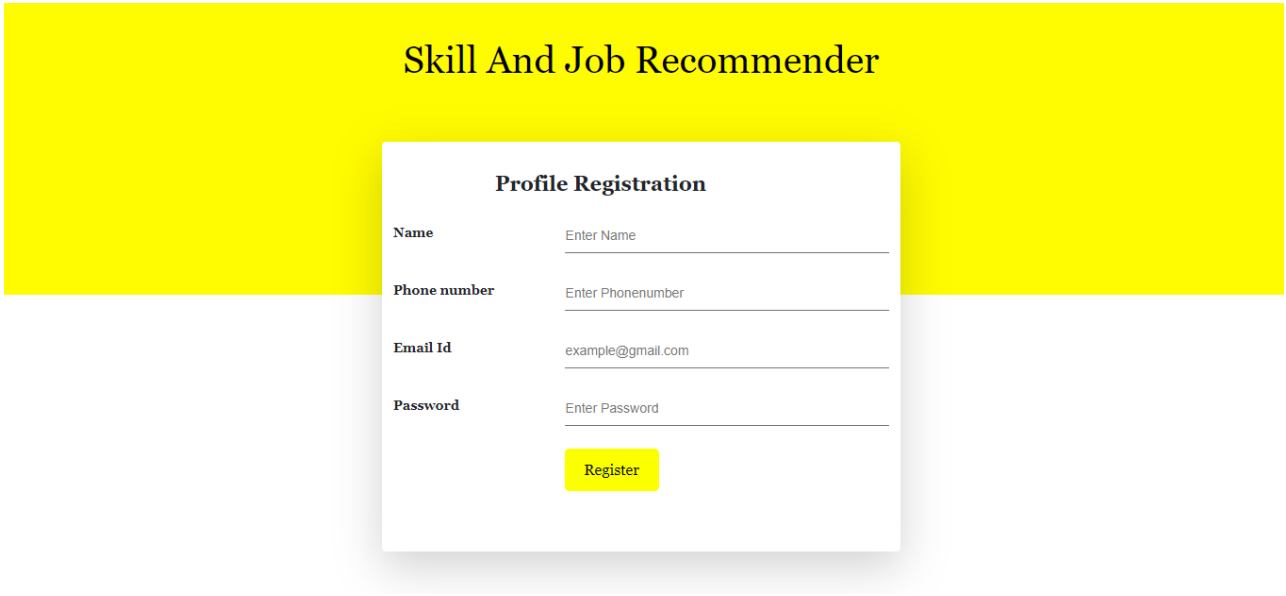
chatbot : it is the User interface used to provide help to find particular search.that it is a user friendly bot to chat and gather information in our website.we have used IBM watson assistant for our chatbot.

OUTPUT:

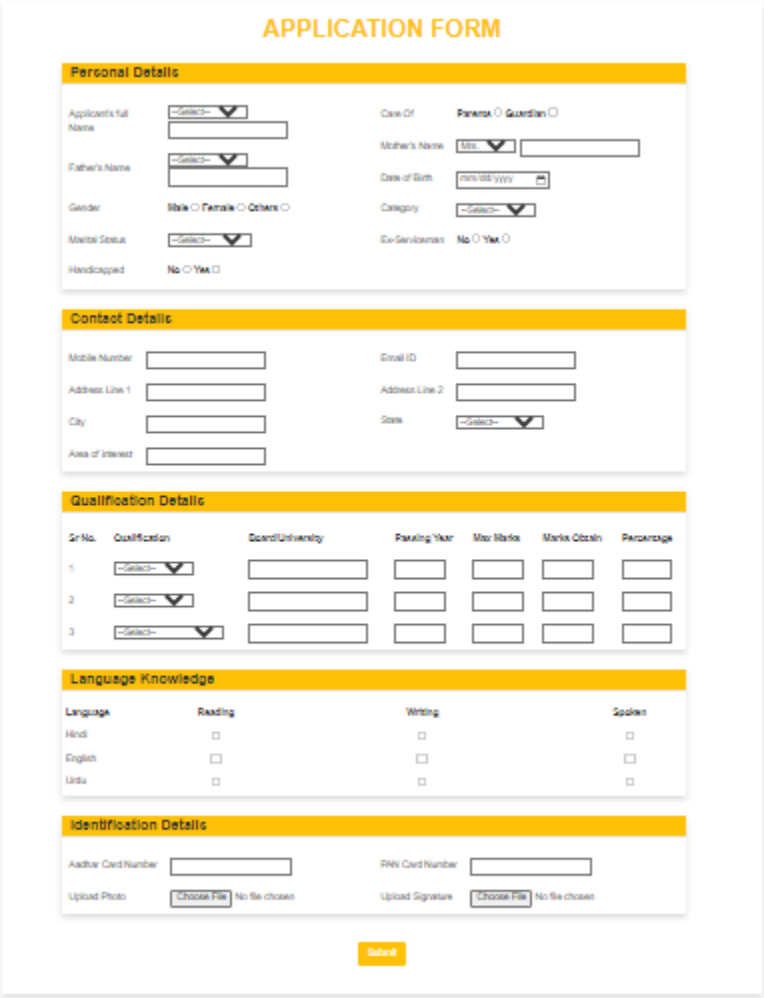
LOGIN PAGE :



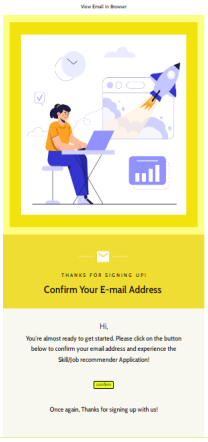
REGISTER PAGE:



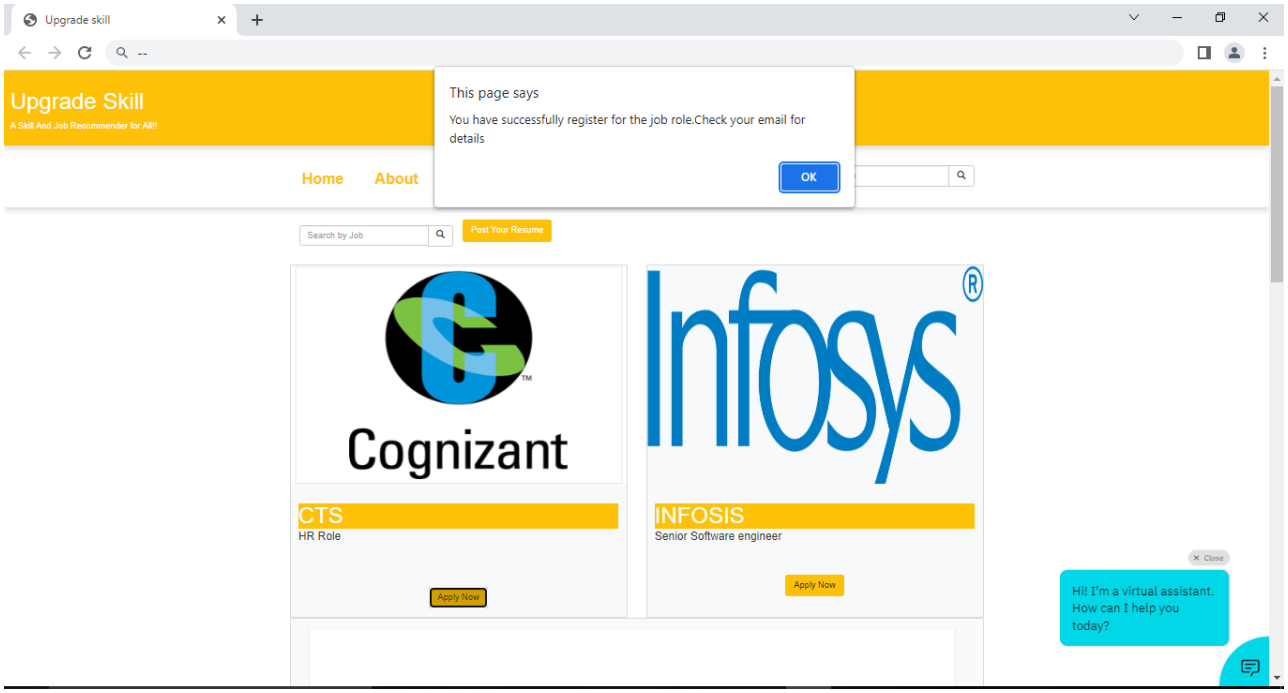
APPLICATION PAGE:



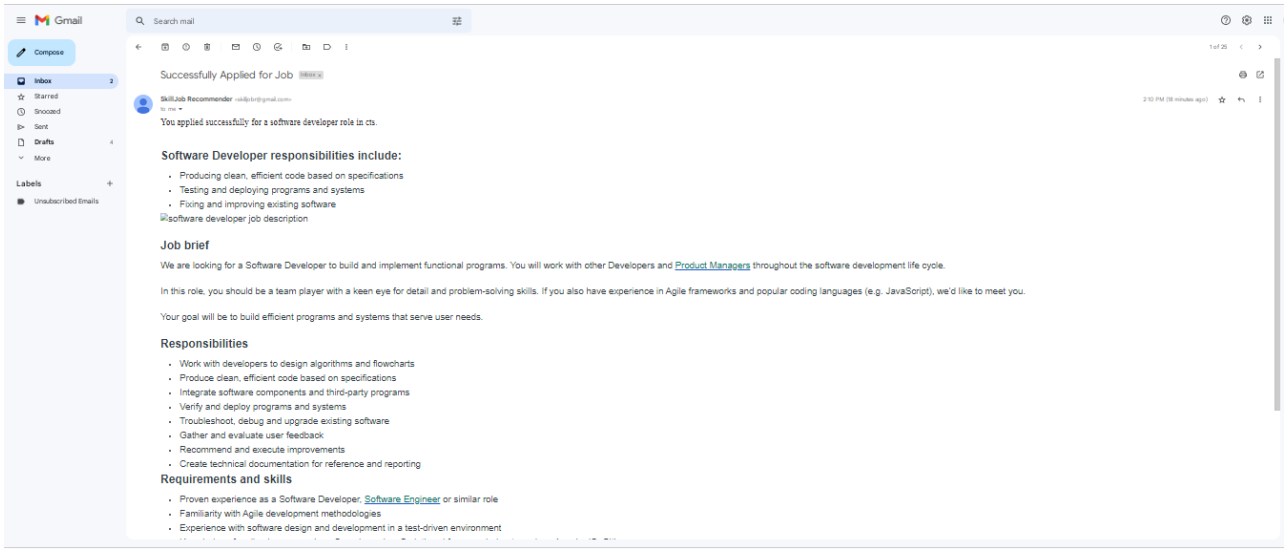
EMAIL VERIFICATION PAGE :



DASHBOARD:



JOB APPLIED VERIFICATION:



8.TESTING

8.1Test Cases

F1 | 11/19/2022

				Date	19-Nov-22									
				Team ID	PNT2022TMID03974									
				Project Name	Project - Skill/Job Recommende									
				Maximum Marks	4 marks									
Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By		
Functional	Login page	Verify that after registration users are navigated to login page	Mail id, Username, Password,Phone number, Pin	1. Open the website and go to register page. 2. Enter details and press register 3. Verify that users are navigated to registration page	https://drive.google.com/file/d/1Ef1snOi59wCYJFmGptiFQBRQVYqSO4M/view?usp=sharing	Users should be navigated to registration page	Working as expected	Pass	Excellent	`N		S S V K Sahit		
UI	Home Page	Verify the UI elements in Login/Signup popup	Username & Password	1. Open the website 2. Enter details and press login 3. Verify that users are notified of login process	https://drive.google.com/file/d/1qW88myufAHRfeQtHjM12CNRd87MFFUO/view?usp=sharing	Users should be notified of login process	Not working	Fail	Trying To Recover	N	BUG-12	Pradeep Narayanan		
Functional	Home page	Verify user is able to log into application with Valid credentials		1. Open the website 2.Enter details and press login 3.Verify that users are logged into website properly	usname: sahitssvk@gmail.com	User should be logged into website properly	Working as expected	Pass	Good	N		S S V K Sahit		
Functional	Home Page	Verify that categories of skills and jobs are shown in homepage		1. Open the website 2. Enter details and press login 3.Verify that categories of are showing jobs shown in homepage		Categories of skills and jobs should be shown in homepage	Working as expected	Pass	Good	N	BUG-14	Akhai Kumar		
Functional	Home page	Verify that jobs are displayed in homepage		1. Open the website 2. Enter details and press login 3.Verify that jobs are displayed in homepage		jobs should be displayed in homepage	Working as expected	Pass	Good	N		Naveen Kumar		
Functional	Home page	Verify that when clicked on jobs it is redirected to correct page		1. Open the website 2. Enter details and press login 3.Verify that when clicked on jobs it is redirected to correct page		When clicked on job link it should be redirected to correct page	Working as expected	Pass	Excellent	N		Pradeep Narayanan		

8.2**User Acceptance Testing:**

2. Defect Analysis

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

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	9	3	2	1	15
Duplicate	0	0	1	2	3
External	2	3	0	1	6
Fixed	8	2	4	14	28
Not Reproduced	0	0	1	0	1
Skipped	0	0	0	1	1
Won't Fix	0	0	0	1	1
Totals	19	14	8	20	55

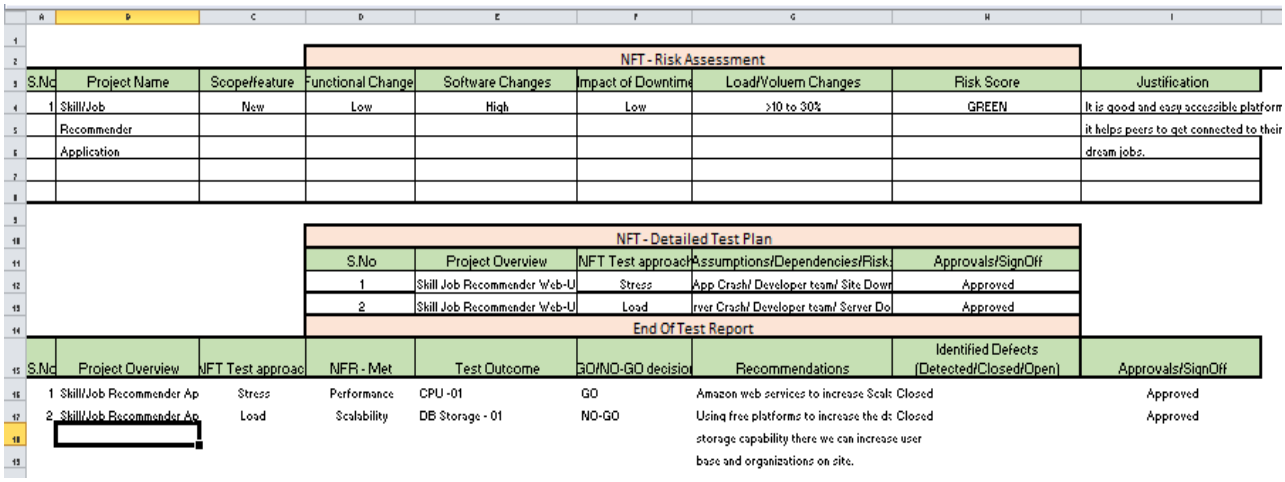
3. Test Case Analysis

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

Section	Total Cases	Not Tested	Fail	Pass
User Application	5	0	0	5
Caretaker Application	20	0	0	20
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9.RESULTS

9.1Performance Metrics:



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Change	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Skill/Job Recommender Application	New	Low	High	Low	>10 to 30%	GREEN	It is good and easy accessible platform it helps peers to get connected to their dream jobs.

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	Assumptions/Dependencies/Risks	Approvals/SignOff
1	Skill Job Recommender Web-U	Stress	App Crash/ Developer team/ Site Down	Approved
2	Skill Job Recommender Web-U	Load	rver Crash/ Developer team/ Server Do	Approved

End Of Test Report

S.No	Project Overview	NFT Test approac	NFR - Met	Test Outcome	GO/NO-GO decisio	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Skill/Job Recommender Ap	Stress	Performance	CPU -01	GO	Amazon web services to increase Scal	Closed	Approved
2	Skill/Job Recommender Ap	Load	Scalability	DB Storage - 01	NO-GO	Using free platforms to increase the da storage capability there we can increase user base and organisations on site.	Closed	Approved

10.ADVANTAGES

- Sending the most appropriate job by integrating with WhatsApp and email.
- Creating resume for job seeker based on his/her choice of template.
- Foreign job opportunities.

DISADVANTAGES

- Offers based on shared characteristics
- Limited slots

11.CONCLUSION:

we have considered the skill/ job recommender system from several perspectives. These include the influence of job competitions, the effect of job availability on the choice of method and filtration, and ethical considerations in job recommender . Furthermore, we branched the web application of hub systems to obtain a better view on how these recommender systems work. we have considered the job recommender system (JRS) literature from several perspectives. These include the influence of data science competitions, the effect of data availability on the choice of method and validation, and ethical considerations in job recommender systems. Furthermore, we branched the large class of hybrid recommender systems to obtain a better view on how these hybrid recommender systems differ. Both this multi-perspective view, and the new taxonomy of hybrid job recommender systems has not been discussed by previous reviews on job recommender systems

12.FUTURE SCOPE

This scheme can be implemented for the future use by creating a web application so that you can use end-end and display the current job openings.An alert system is used for sending the notification of the jobs depending upon the filtration process.A new job search API can be used for new job openings in the market from the webpage directly and then also connecting through social media.Data collection and preprocessing followed by the unification of the database also can be implemented.

13. Appendix:

source code link:

https://github.com/IBM-EPBL/IBM-Project-21946-1659798818

Demo Video Link:

https://drive.google.com/drive/folders/1Me05tI89yQ7rAKDAWop7bJumT8y-cYOW?usp=share_link