Professional Readiness For Innovative,

Project Name: Skill/Job Recommender Application

Team Id: PNT2022TMID22285

Team Leader:BarathRaj.T

Team Member 1:Barath.M

Team Member 2: Shankara Narayanan.B

Team Member 3: Prithivirajkumar.K

1. INTRODUCTION

1.1 Project Overview

1.2 Purpose

2. LITERATURE SURVEY

2.1 Existing problem

2.2 References

2.3 Problem Statement Definition

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas
3.2 Ideation & Brainstorming
3.3 Proposed Solution
3.4 Problem Solution fit

4. REQUIREMENT ANALYSIS

4.1 Functional requirement
4.2 Non-Functional requirements

5. PROJECT DESIGN

5.1 Data Flow Diagrams
5.2 Solution & 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

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

7.1 Feature 1

7.2 Feature 2

7.3 Database Schema (if Applicable)

8. TESTING

8.1 Test Cases

8.2 User Acceptance Testing

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

11.CONCLUSION

12. FUTURE SCOPE

13. APPENDIX

Source Code

GitHub & Project Demo Lin

1. INTRODUCTION

The increasing usage of Internet has heightened the need for online job hun ng. The key problem is that most of job-hun ng websites just display recruitment informa on to website viewers. Students have to retrieve among all the informa on to find jobs they want to apply. The whole procedure is tedious and inefficient. In addi on, many E-commerce websites, the most general applica on of recommenda on algorithms, uses collabora ve filtering algorithm without considering user's resume and item's proper es in this case, that means students' resume and details of recrui ng informa on. So we proposed an improved algorithm based on item-based collabora ve filtering. The aim of the present paper is to give an effec ve method of recommenda on for online job hun ng. We hope to offering students a personalized service that can help them

find ideal jobs quickly and conveniently. In this paper, we present a survey of exis ng recommenda on approaches that have been used for building the personalized recommenda on systems for job seekers as well as recruiters. Also we have iden fied the challenges in building a job recruitment system as compared to recommenda on systems in other domain. Even so, the sparsity of user profile can be obstruc ve, further studies on filling users' preference matrix with implicit behaviors will be summarized in our next study.

1.1 PROJECT OVERVIEW

In the last years, job recommender systems have become popular since they successfully reduce informa on overload by genera ng personal-ized job sugges ons. Although in the literature exists a variety of techniques and strategies used as part of job recommender systems, most of them fail to recommending job vacancies that fit properly to the job seekers profiles. Thus, the contribu ons of this work are threefold, we: i) made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites; ii) put forward the proposal of a framework for job recommenda on based on professional skills of job seekers; and iii) carried out an evalua on to quan fy empirically the recommenda on abili es of two state-ofthe-art methods, considering different configura ons, within the proposed framework.

Keywords: “Job matching, job seeking, job search, job recommender systems, person-job fit, LinkedIn, word embedding”

We thus present a general panorama of job recommenda on task aiming to facilitate research and real-world applica on design regarding this important issue. From the last two decades internet based recrui ng pla orms have become a primary channel in most companies for recrui ng talents. Such portals decrease the adver sement cost, but they suffer from informa on overload problem. Job portals using tradi onal informa on retrieval techniques such as Boolean search methods are typically using simple word matching algorithms. The main issue of these portals is their inability to understand the complexity of matching between candidates' desires and organiza ons' requirements. Hence, a vast amount of

deserving candidates misses the opportunity to get an appropriate job. The recent recommender systems have achieved success in e-commerce applica ons. In order to improve the func onality of e-recruitment process, many recommenda on systems approaches have been proposed.

1.2 PURPOSE

With an increasing number of cash-rich, stable, and promising technical companies/startups on the web which are in much demand right now, many candidates want to apply and work for these companies. They tend to miss out on these postings because there is an ocean of existing systems that list millions of jobs which are generally not relevant at all to the users. There is an abundance of choices and not much streamlining. On the basis of the actual skills or interests of an individual, job seekers often find themselves unable to find the appropriate employment for themselves. This system, therefore, approaches the idea from a data point of view, emphasizing more on the quality of the data than the quantity.

2. LITERATURE SURVEY

1) Dynamic User Profile-Based Job Recommender System

Approach/Methodology:

The basic features are extracted from the job seeker's profile.The profile might get out-dated when the user does not update it in a timely manner. Based on the behaviors of the job applicant and the previous jobs which he/she applied for, the dynamic features are extracted which is an updated version of basic features. So, the system makes a statistic at regular intervals, to generate the dynamic basic features. With the increasing number of applied jobs, the number of extracted features becomes greater. Information gain is calculated for each feature. More the information gain for a feature, the more important the feature is. The dynamic recommendation system works as follows: Initiallyfor solving the cold-start problem, the user based collaborative filtering algorithm is applied to generate theinitial recommendation jobs. After having the initial recommendations, the system provides the recommendations to the job seeker and records his behavior. The interested and uninterested jobs set is generated by analyzing the job applicant's behavior. Thus, the interested job set helps in extending and updating the user profile. Thus, the new basic features are used to calculate the similarity between the job applicantand job vacancies. So, new recommendations will be made available for the job applicant.

Pros:

Job applicants do not update their profile in a timely manner. This system aims at updating and extending the user profile dynamically based on the historical appliedjobs and the behavior of job applicants.

Cons:

Besides the time and the dimensionality of features, there are other factors that affect the dynamic job recommendation system. The context formed in the peak season and the off season has an influence on the job desire of a job applicant. The drawbackof this system is that it does not take these other factors into consideration.

2) Temporal Learningand Sequence Modelingfor a Job Recommender System

Approach/Methodology:

The approachcombines temporal learningwith sequence modelingto capture complex user-itemactivity patterns to improve job recommendations. It is a timebased ranking model applied to historical observations and a hybrid matrix factorization over time reweighted interactions. Second,it exploitssequence properties in user-items activities and develops a RNN-based recommendation model.

Pros:

TheModel is comparedto two baseline models: randomized score (Rand) and recency-based sorting (TSort)

3) Collaborative Job Prediction based on Naive Bayes Classifier using Python Platform

Approach/Methodology:

The proposed method includes implementing a recommendation system based on the collaborative filtering technique for job portals. The system is designed to suggest the jobs to the user depending upon his profileand by calculating a

similarity index using Euclidean distance of two skill sets and then ranking them accordingto their naïve Bayes algorithm.

Pros:

It has small computational overhead compared to Machine learningmodels.

Cons:

Susceptible to cold-start problem

4) Combining content-based and collabora ve filteringfor job recommenda on system: A cost-sensitive Statistical Relational Learning approach

Approach/Methodology:

Developed a hybrid content-based filtering and collaborative filtering approach. The approach adapted a successful Statistical Relational Learning algorithm for learning features and weights and is capable of handling different costs for falsepositives and false negatives. The hybrid recommendation system is constructed by learning the Relational Dependency Network using a state-of-theart learning approach–Relational Functional Gradient Boosting.

Pros:

Prevents the necessity for exhaustive featureengineering or pre-clustering and provides a robust way to solve the cold-start problem.

Cons:

Markov Logic Networks with Alchemy2 failaue to large amounts of data.

5) A CombinedRepresentation Learning Approachfor Better Job and SkillRecommendation

Approach/Methodology:

The proposedsolution is representation learning based that leverages information of three graphs in order to representeach job and skill into a shared lowdimensional vector space for solving the job recommendation task from the historical job data:

Pros:

The proposed embedding methodology consistently outperforms three state-of-the art methods in terms of job recommendation task, which improvesHR, NDCG, and pair-wise AUC by 3.4%, 6.7%, 1.2%, respectively.

Cons:

The proposed representa on learning framework is transduc ve, i.e., it learnsrepresenta on vectors of jobs and skills that are available in the input graphs and new job tles and skills are not suggested

6) JobRecommendation based on Job Seeker Skills: An Empirical

StudyApproach/Methodology:

The skills are extracted from the job seeker profiles using various text processing techniques. Job recommendation is performed using TF-IDF and four different configurations of Word2Vec over a datasetof job seeker profiles and job vacancies. A group of nearest job offers based on distance to the job seeker's profile is selected (job matching). In the case of TF-IDF representation, cosine distance is used, while for word embeddings, the new Word Mover's

Distance(WMD) is used. Once retrieved the top ”k” job offers for the profile, they are sorted in descending order based on the inverse of this distance(ranking)

Pros:

Personalized job recommendation is done based on the job seeker’sprofile. Recommendations based on other data like query based on keywords related to the job vacancy that the job seeker is looking for, etc. are less accuratethan personalized job recommendations.

Cons:

TF-IDF computes document similarity directly in word-count space. It makes no use of semanticsimilarities between words.

Word2Vec modelhas an inability to handleunknown or out of vocabulary words.

2.1 EXISTING SYSTEM

Existing system is not very efficient , it does not benefit the user in maximum way, so the proposed system uses ibm cloud services like db2, Watson virtual assistant , cluster , kubernetes and docker for containerization of the application.

2.2 REFERENCES:

- T Al-Otaibi and Mourad Ykhlef. “A survey of job recommender systems”. In: International Journal of the Physical Sciences 7.29 (2012), pp. 5127–5142. issn: 19921950. doi: 10.5897/IJPS12. 482

- N Deniz, A Noyan, and O G Ertosun. “Linking Person-job Fit to Job Stress: The Mediating Effect of Perceived Person-organization Fit”. In: Procedia - Social and Behavioral Sciences 207 (2015), pp. 369– 376.

- M Diaby, E Viennet, and T Launay. “Toward the next generation of recruitment tools: An online social network-based job recommender system”.

In: Proc. of the 2013 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, ASONAM 2013 (2013), pp. 821–828. doi: 10. 1145/2492517.2500266.

- M Diaby and E Viennet. “Taxonomy-based job recommender systems on Facebook and LinkedIn profiles”. In: Proc. of Int. Conf. on Research Challenges in Information Science (2014), pp. 1–6. issn: 21511357. doi: 10.1109/RCIS.2014.6861048.

- M Kusner et al. “From word embeddings to document distances”. In: Proc. of the 32nd Int. Conf. on Machine Learning, ICML’15. 2015, pp. 957–966.

- T Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: Proc. of the 26th Int. Conf. on Neural Information Processing Systems - Volume 2. NIPS’13. Lake Tahoe, Nevada, 2013, pp. 3111–3119. url: http://dl.acm.org/citation.cfm?id=2999792. 2999959.

- T Mikolov et al. “Efficient estimation of word representations in vector space”. In: arXiv preprint arXiv:1301.3781 (2013).

- G Salton and C Buckley. “Term-weighting approaches in automatic text retrieval”. In: Information Processing and Management 24.5 (1988), pp. 513–523. issn: 0306-4573. doi: https://doi.org/10. 1016/0306- 4573(88)90021- 0. url: http://www.sciencedirect.com/science/article/pii/ 030645738890021

2.3 PROBLEM STATEMENT DEFINITION

In the last years, job recommender systems have become popular since they successfully reduce informa on overload by genera ng personalized job sugges ons. Although in the literature exists a variety of techniques and strategies used as part of job recommender systems, most of them fail to recommend job vacancies that fit properly to the jobseekers profiles. Thus, the contribu ons of this work are threefold, made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites, put forward the proposal of a framework for job recommenda on based on professional skills of job seekers, and carried out an evalua on to quan fy empirically the recommenda on abili es of two state-of-the-art methods, considering different

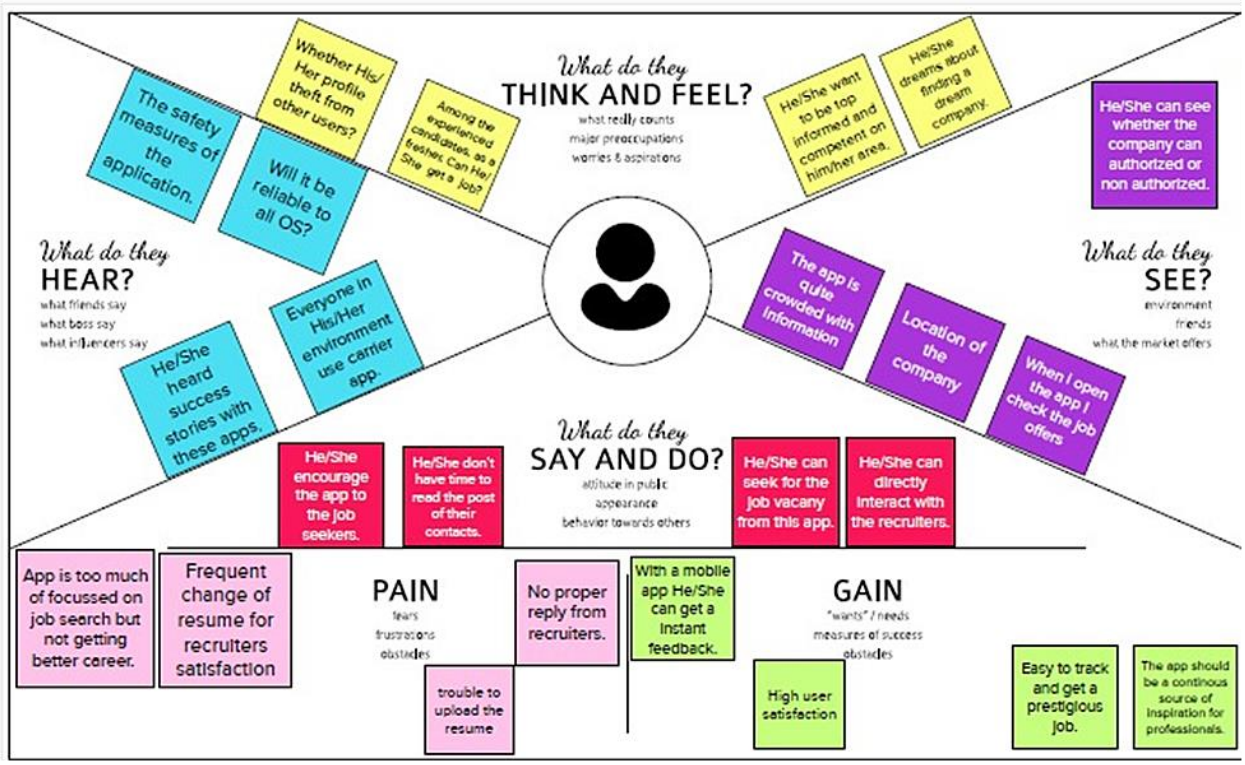
configura ons, within the proposed framework. Thus present a general panorama of job recommenda on task aiming to facilitate research and real-world applica on design regarding this important issue. Job matching, job seeking, job search, job recommender systems.

Proposed a framework for job recommenda on task. This framework facilitates the understanding of job recommenda on process as well as it allows the use of a variety of text processing and recommenda on methods according to the preferences of the job recommender system designer. Moreover, we also contribute making publicly available a new dataset containing job seekers profiles and job vacancies. Future direc ons of our work will focus on performing a more exhaus ve evalua on considering a greater amount of methods and data as well as a comprehensive evalua on of the impact of each professional skill of a job seeker.

3) IDEATION & PROPOSEDSOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by **Dave Gray** and has gained much popularity within the agile community.

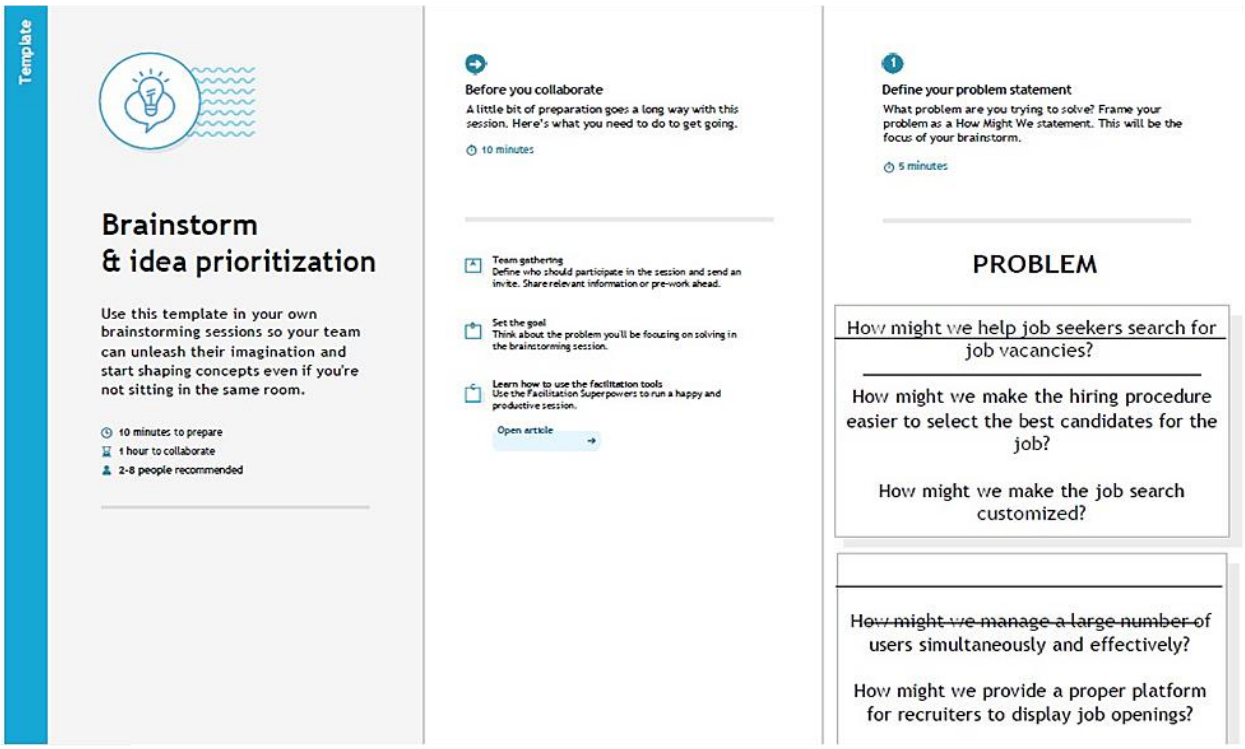


3.2 IDEATION & BRAINSTORMING

Ideation refers to the whole creative process of coming up with and communicating new ideas. It can take many different forms, from coming up with a totally new idea to combining multiple existing ideas to create a new process or organizational system. Ideation is similar to a practice known as brainstorming.

Brainstorming sets the stage for the rest of the ideation process, so it's something you should approach with deliberate strategy. A typical brainstorming session involves one or more people directing their thoughts towards a particular problem or issue.

Step 1


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
Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.
Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.
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 job seekers search for job vacancies?
How might we make the hiring procedure easier to select the best candidates for the job?
How might we make the job search customized?
How might we manage a large number of users simultaneously and effectively?
How might we provide a proper platform for recruiters to display job openings?

Brainstorming.pdf

1 / 1 25%

ement

ve? Frame your
nent. This will be the

le over the
base is
uge set of
arch.

storming
ductive session

Encourage wild ideas.

2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

BARATH M

First I would analyse choice user.

Provided or suggested only trusted jobs.

User privacy guarded.

BARATH RAJ T

We should prefer a worthy job.

The jobs and skills provided are really helpful.

Create the information should for when jobs are available.

PRITHVI RAJ KUMAR K

The user able to determine skill.

Getting user choice.

It provides confidence for the user.

SHANKARANARAYANAN B

Analysing the personal skills.

Watching the progress of other people.

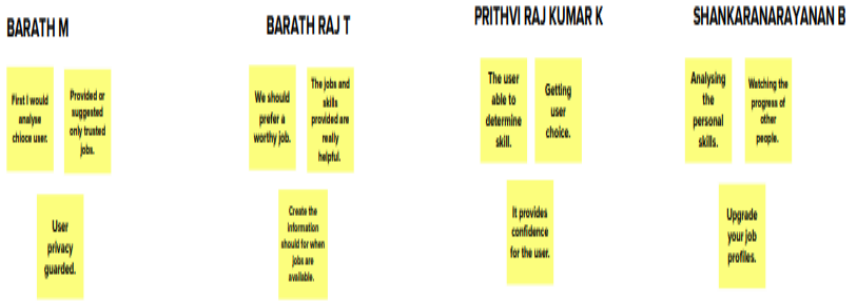
Upgrade your job profiles.

3

Group ideas

Take turns sharing your ideas
sticky notes have been group
bigger than six sticky notes, t

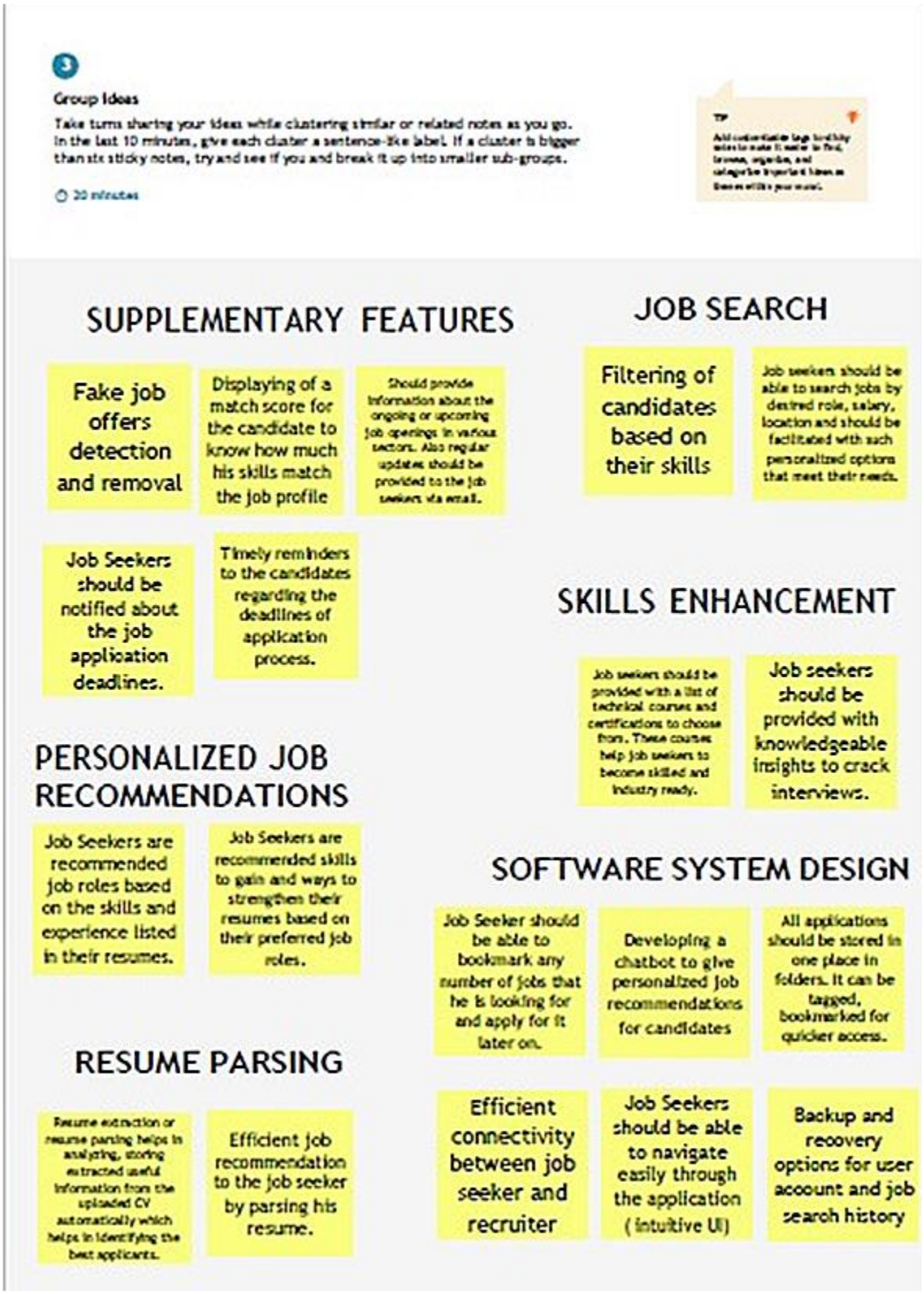
20 minutes

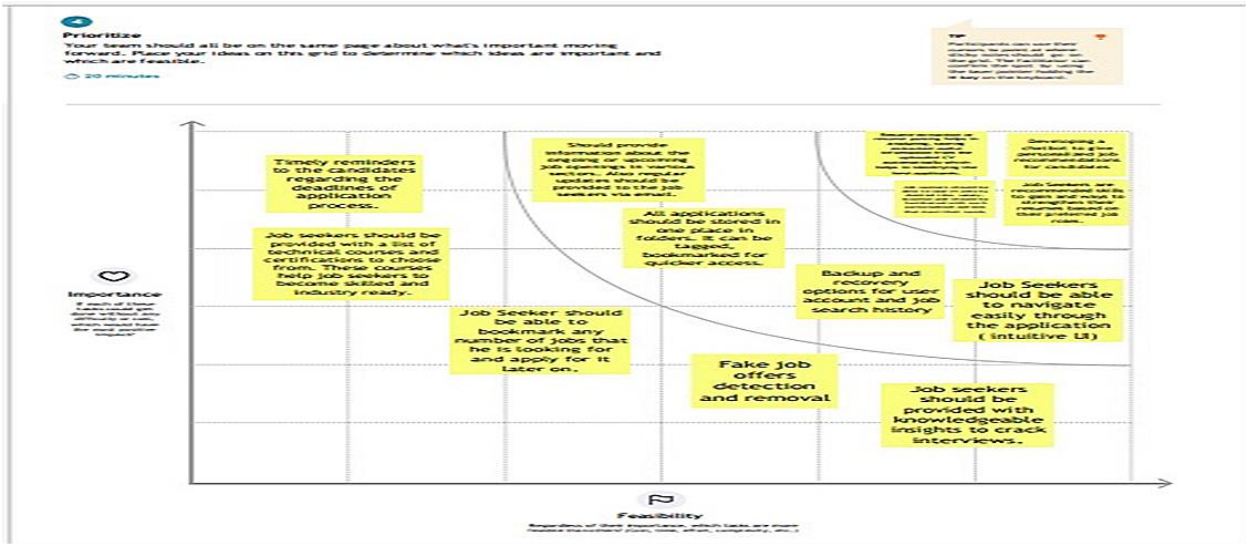


Security

User privacy guarded

User
ar
pass
secu


Group Ideas
Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.
20 minutes
SUPPLEMENTARY FEATURES
Fake job offers detection and removal
Displaying of a match score for the candidate to know how much his skills match the job profile
Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
Job Seekers should be notified about the job application deadlines.
Timely reminders to the candidates regarding the deadlines of application process.
JOB SEARCH
Filtering of candidates based on their skills
Job seekers should be able to search jobs by desired role, salary, location and should be facilitated with such personalized options that meet their needs.
SKILLS ENHANCEMENT
Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
Job seekers should be provided with knowledgeable insights to crack interviews.
PERSONALIZED JOB RECOMMENDATIONS
Job Seekers are recommended job roles based on the skills and experience listed in their resumes.
Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.
SOFTWARE SYSTEM DESIGN
Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
Developing a chatbot to give personalized job recommendations for candidates
All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
RESUME PARSING
Resume extraction or resume parsing helps in analyzing, storing extracted useful information from the uploaded CV automatically which helps in identifying the best applicants.
Efficient job recommendation to the job seeker by parsing his resume.
Efficient connectivity between job seeker and recruiter
Job Seekers should be able to navigate easily through the application (intuitive UI)
Backup and recovery options for user account and job search history



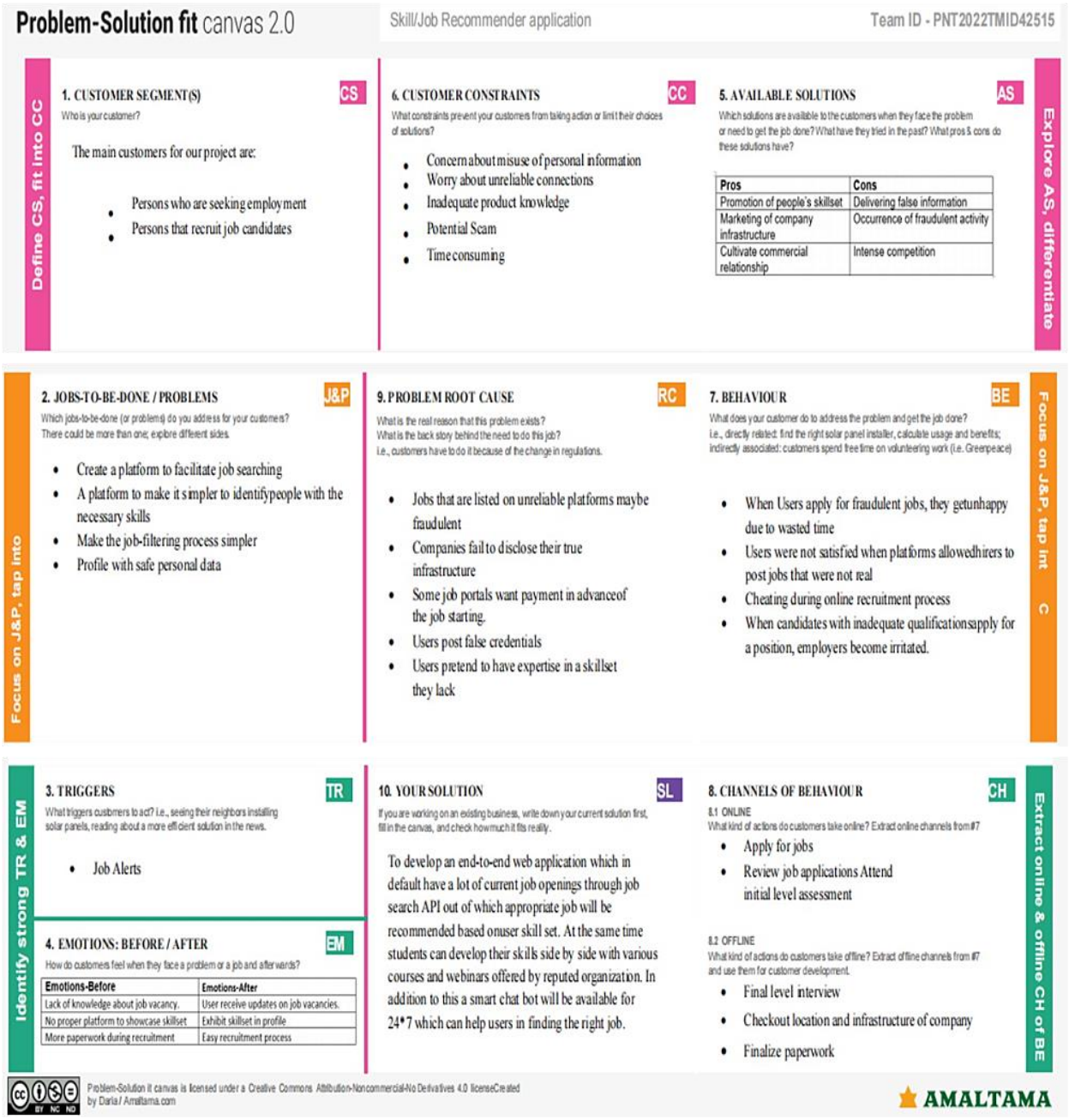
3.3 PROPOSED SOLUTION:

Proposed Solution Template:

Project team shall fill the following information in proposed solution template.

S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	➢ Premium policy is an issue to the users. ➢ Look for field based jobs as searching for fields as a whole is time-consuming. ➢ Estimating salaries based on technical skills.
2.	Idea / Solution description	➢ Free access to every users. ➢ Filtering job by it's categories. ➢ Salary calculator for the estimation of the pay.
3.	Novelty / Uniqueness	➢ Refinement of the job fields. ➢ Earnings estimator based on knowledge of users.
4.	Social Impact / Customer Satisfaction	➢ Open doors for every users as there is free access. ➢ Users stay up to date of the offers.
5.	Business Model (Revenue Model)	➢ Advertising about the platform. ➢ Regularly updating the new technologies and jobs offers.
6.	Scalability of the Solution	➢ Scalable at Professional Training and Coaching. ➢ Scalability in finding more parent-friendly environment. ➢ Creating a positive culture is the main cause in maximizing the productivity.

3.4 PROBLEM SOLUTION FIT:



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration via Form And via Gmail
FR-2	User Confirmation	Confirmation through Email That is through OTP
FR-3	Chat Bot	A Chat Bot will be there in website to solve user queries and problems related to applying a job, search for a job and much more.
FR-4	User Login	Login through Form Login through Gmail
FR-5	User Search	Exploration of Jobs based on job filters and skill recommendations.
FR-6	User Profile	Updation of the user profile through the login credentials
FR-7	User Acceptance	Confirmation of the Job.

2 4.NON FUNCTIONAL REQUIREMENT:

Non-functional Requirements:

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

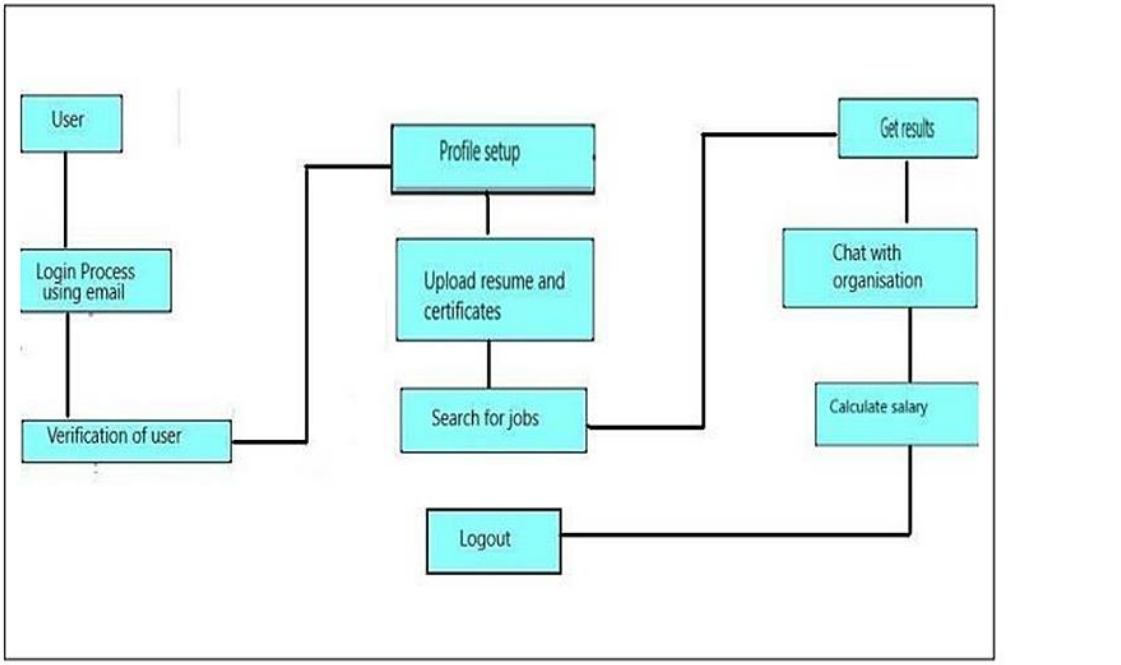
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	This application can be used by the job seekers to login and search for the job based on her Skills set.
NFR-2	**Security**	This application is secure with separate login for Job Seekers as well as Job Recruiters.
NFR-3	**Reliability**	This application is open-source and feel free to use, without need to pay anything. The enormous job openings will be provided to all the job seekers without any limitation.
NFR-4	**Performance**	The performance of this application is quicker response and takes lesser time to do any process.
NFR-5	**Availability**	This application provides job offers and recommends Skills for a Particular Job openings.
NFR-6	**Scalability**	The Response time of the application is quite faster compared to any other application.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

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 enter and leaves the system, what changes the information, and where data is stored.



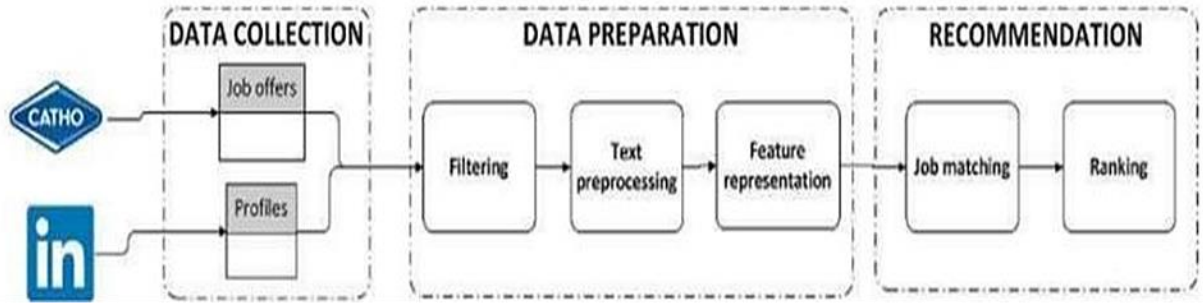
5.2 SOLUTION & TECHNICAL ARCHITECTURE

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Example - Solution Architecture Diagram:



5.3 USER STORIES

User Stories

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail.		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 signing in.	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Access	USN-6	As a user, I can setup a profile, and basic details by signing in.			
		USN-7	As a user, I will upload my resume, certificates, and other requirements.	I can perform several task in the application	Medium	Sprint-1
Customer Care Executive	Chat bot	USN-8	As a user, I can seek guidance from the customer care executive.		High	Sprint-1
Administrator	DBMS	USN-9	As a administrator, I can keep the applications of your organization relies on running.	I can perform various modifications in the applications.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers, research publications etc	2 SEPTEMBER 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	10 SEPTEMBER 2022
Ideation	Prioritise a top 3 ideas based on feasibility and Importance	17 SEPTEMBER 2022
Proposed Solution	Solution include novelty, feasibility, business model, social impact and scalability of solution	24 SEPTEMBER 2022
Problem Solution Fit	Solution fit document	29 SEPTEMBER 2022
Solution Architecture	Solution Architecture	1 October 2022
Customer Journey	To Understand User Interactions and experiences with application	8 October 2022
Functional Requirement	Prepare functional Requirement	14 October 2022
Data flow Diagrams	Data flow diagram	15 October 2022
Technology Architecture	Technology Architecture diagram	16 October 2022
Milestone & sprint delivery plan	Activity what we done &further plans	21 October 2022
Project Development- Delivery of sprint 1,2,3 &4	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

6.2 SPRINT DELIVERY SCHEDULE

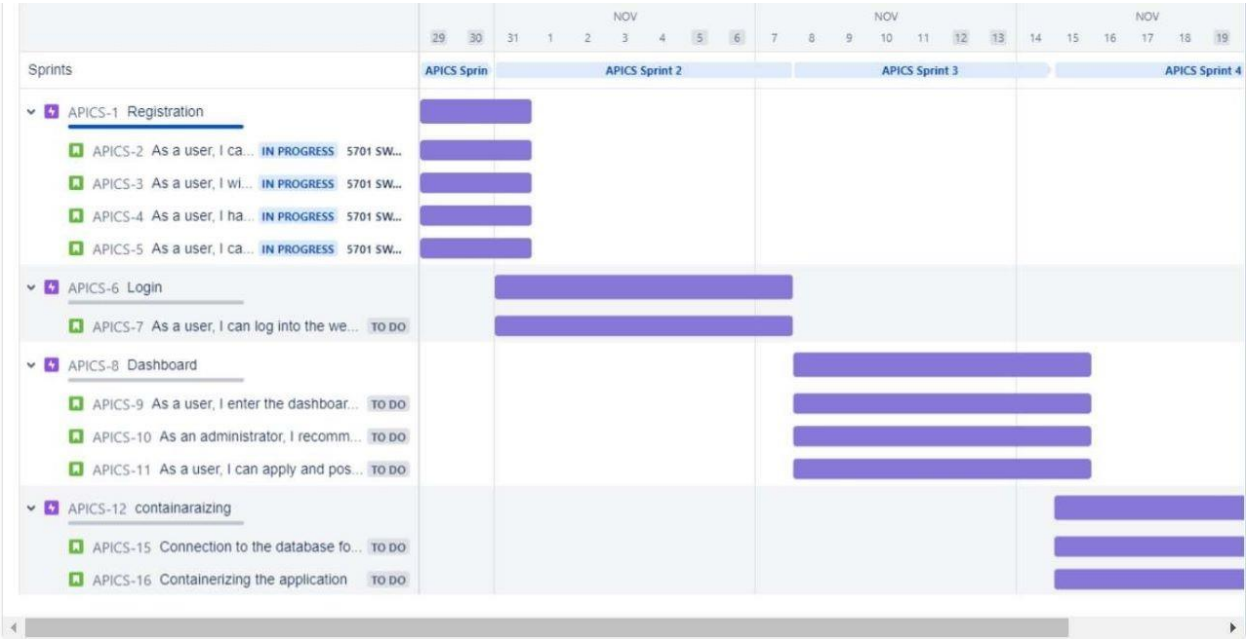
SPRINT DELIVERY SCHEDULE

SPRINT	TASK	MEMBERS
SPRINT 1	Create Registration page ,login page , Job search portal , job apply portal in flask	**BarathRaj.t** **Barath.M**
SPRINT 2	Connect application to ibmdb2	**BarathRaj.T** **Shankara Narayanan**
SPRINT 3	Integrate ibm Watsonassisstant	**Prithivi Raj Kumar** **BarathRaj .T**
SPRINT 4	Containerize the app and Deploy the application in ibm cloud	**Barathraj.T** **Barath.M** **Shankara Narayanan** **Prithivi raj kumar**

6.3 REPORTS FROM JIRA

Average Age Report. Created vs Resolved Issues Report. Pie Chart Report.

Recently Created Issues Report. Resolu on Time Report. Single Level Group By Report. Time Since Issues Report. Time Tracking Report.



7. CODING & SOLUTIONING

FEATURE 1:

App Market

This is one of the feature of our applica on F-ing Jobs which provides companies job details for end users

```
from flask import Blueprint, jsonify, request

from backend import conn

from backend.auth_middleware import token_required

import ibm_db

user = Blueprint("user", _name_)

@user.route("/skills", methods=["GET", "POST", "DELETE"])

@token_required
```

```
def manage_skills(current_user):
# Get user_id of current user
user_id = current_user['USER_ID']
   # Handle GET request     if
request.method == 'GET':
       skills = []
       sql = f"select name from skills where user_id={user_id}"
stmt = ibm_db.prepare(conn, sql)
ibm_db.execute(stmt)
```

```
        dict = ibm_db.fetch_assoc(stmt)
        # Iterate over all the results and append skills to the
array         while dict != False:
skills.append(dict['NAME'])             dict =
ibm_db.fetch_assoc(stmt)

        return jsonify({"skills": skills}), 200
# Get the skills from the request     if
not ('skills' in request.json):
        return jsonify({"error": f"All feilds are required!"}), 409
skills = request.json['skills']
   # If no skills are provided then return empty array
if skills == []:
        return jsonify({"skills": []}), 200
# Handle POST request     if
request.method == "POST":
        # Prepare the SQL statement to insert mul ple rows
values = ''         for i in range(len(skills)):             if i == 0:
values += 'values'             values +=
f"('{skills[i]}',{user_id})"
```

```
        if i != len(skills)-1:
values += ','
    sql = f"insert into skills(name,user_id) {values}"
stmt = ibm_db.prepare(conn, sql)        status =
ibm_db.execute(stmt)
    if status:
        return jsonify({"message": "Updated skills successfully!"}), 200
else:
        jsonify({"error": "Something went wrong!!"}), 409
  # Handle DELETE request     if
request.method == 'DELETE':
values = ""
    for i in range(len(skills)):
values += f"'{skills[i]}'"            if
i != len(skills)-1:
values += ','
    sql = f"delete from skills where name in
({values})"        stmt = ibm_db.prepare(conn, sql)
status = ibm_db.execute(stmt)
    if status:
```

```
        return jsonify({"message": "Deleted skills successfully!"}), 200
else:

jsonify({"error": "Something went wrong!!"}), 409
@user.route('/profile', methods=["POST"])
@token_required
def update_user_info(current_user):
user_id = current_user['USER_ID']
update_fields = ['name', 'phone_number']
for feild in update_fields:        if not (feild in
request.json):
        return jsonify({"error": f"All feilds are required!"}), 409
name = request.json['name']
   phone_number = request.json['phone_number']
   sql = f"update users set
name='{name}',phone_number='{phone_number}' where user_id={user_id}"
stmt = ibm_db.prepare(conn, sql)     status = ibm_db.execute(stmt)
   if status:
      return jsonify({"name": name, "phone_number": phone_number}), 200
else:
      jsonify({"error": "Something went wrong!!"}), 409
```

FEATURE:2

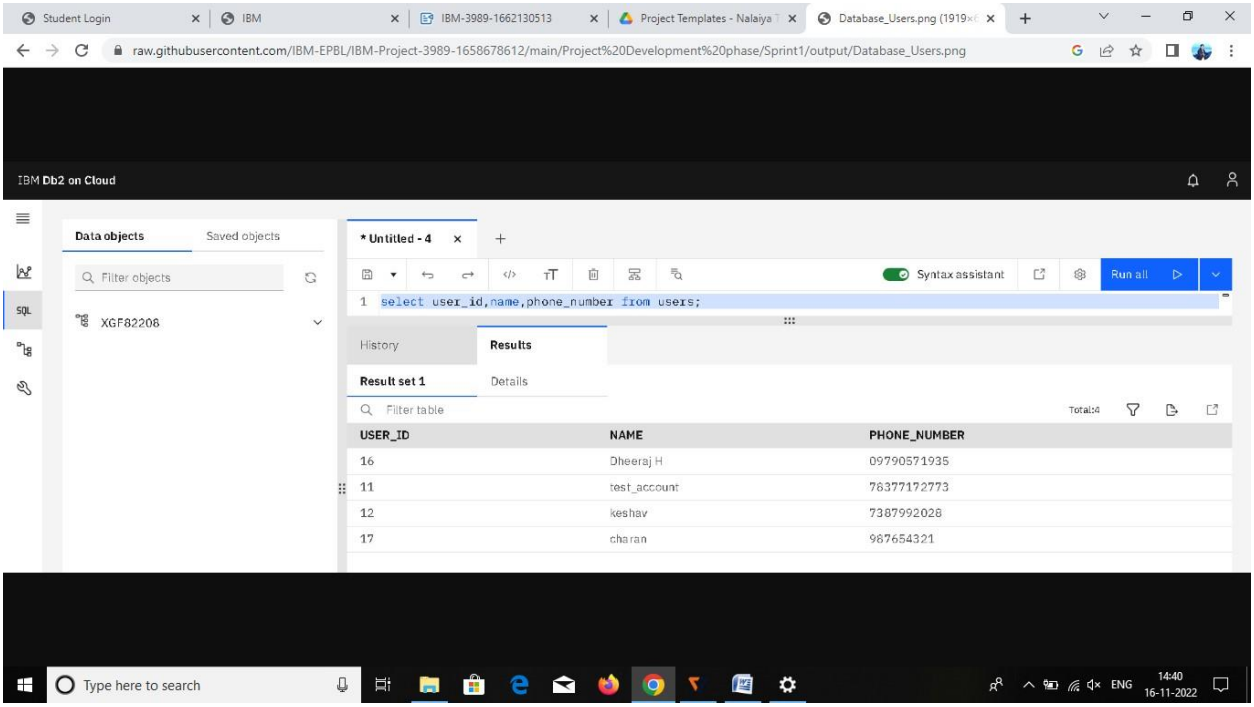
INTEGRATING CHATBOT TO HTML PAGE

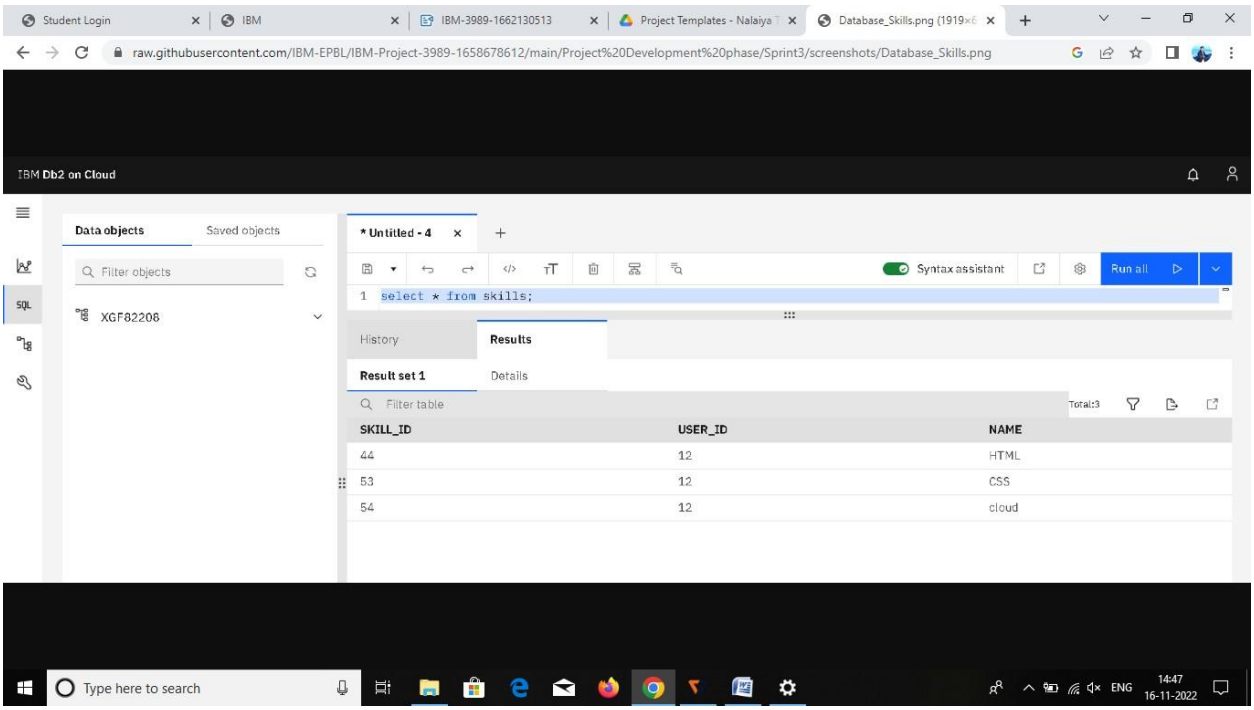
```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta h p-equiv="X-UA-Compa ble" content="IE=edge">

<meta name="viewport" content="width=device-width, ini al-scale=1.0">

< tle>Document</ tle>

</head>

<body>

<h1>My Chatbot</h1>

<blockquote>Click the bo om right corner to chat</blockquote>

<script>

window.watsonAssistantChatOp ons = {

integra onID: "01ca5fe5-3f42-4a97-8965-332afedd97be", // The ID of this
integra on.

region: "au-syd", // The region your integra on is hosted in.

serviceInstanceID: "5683f375-e95c-4fa1-8471-5b76177675c2", // The ID of your

service instance.

onLoad: func on(instance) { instance.render(); }
```

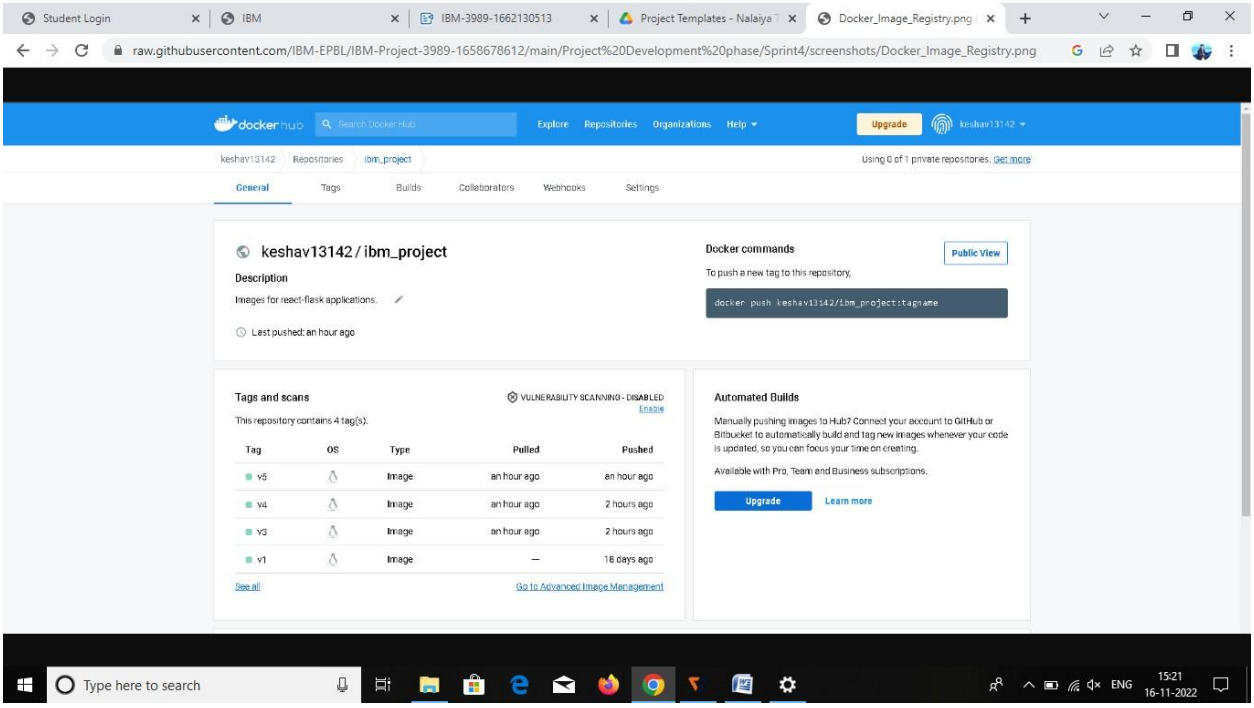
```
};
setTimeout(func on(){
const t=document.createElement('script');
t.src="h ps://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOp ons.clientVersion || 'latest') +
"/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script>
</body>
</html>
```

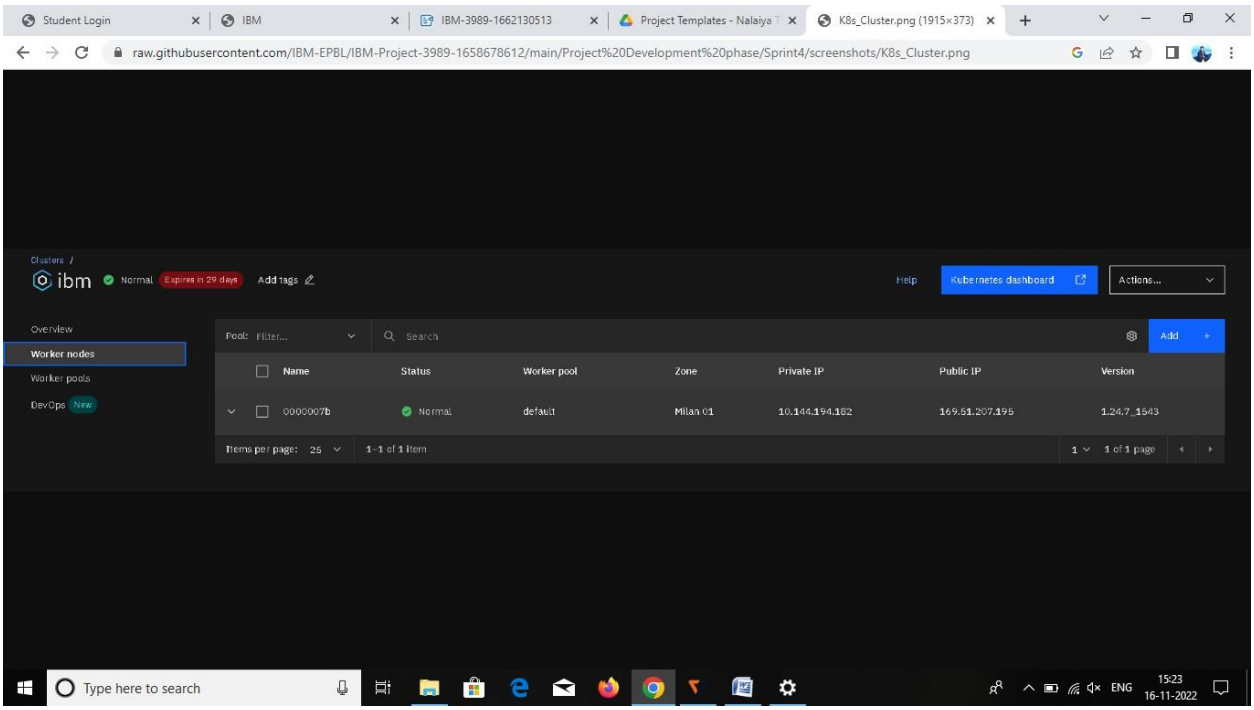
7.3 Database Schema:

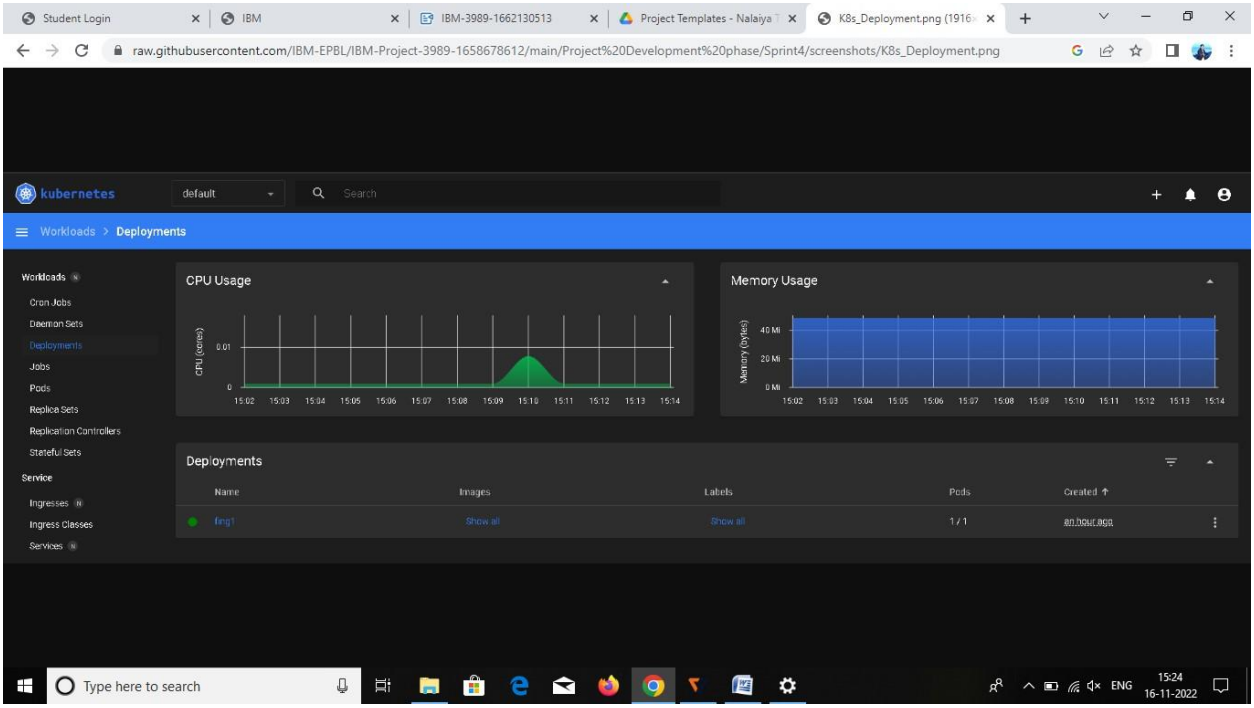
We user IBM DB2 for our database, below are the tables we used with the parameters given.

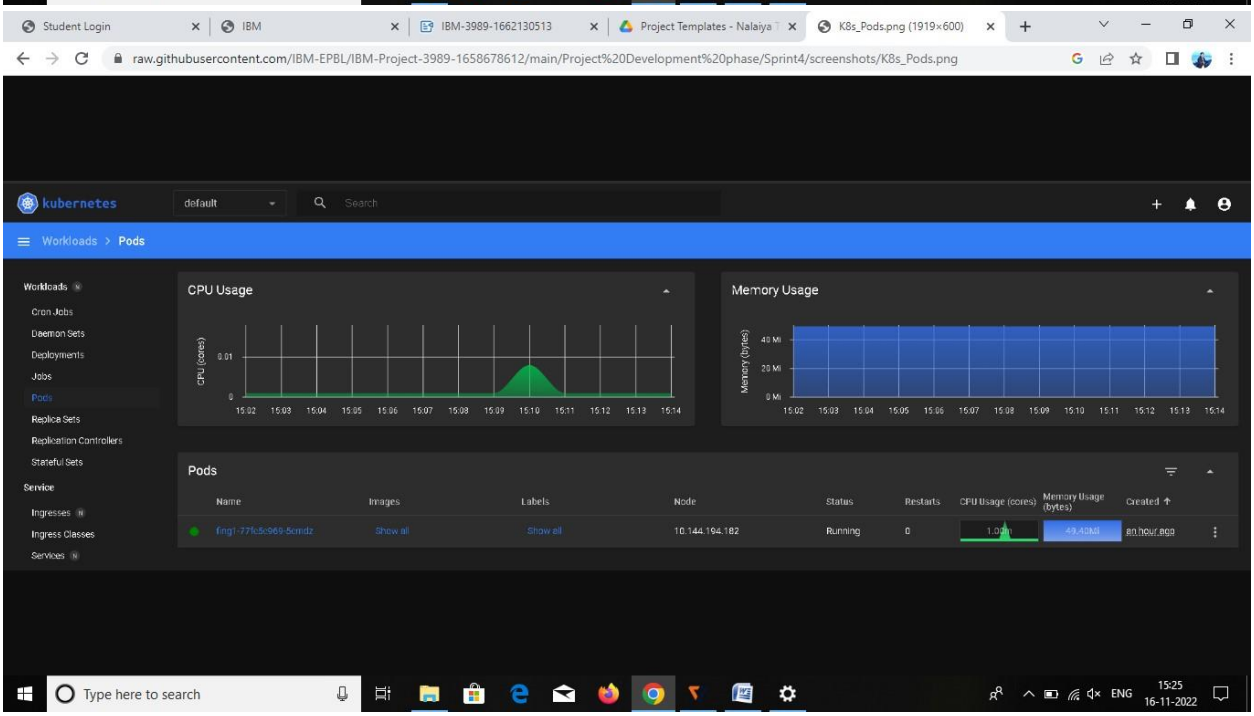

Student Login
IBM
IBM-3989-1662130513
Project Templates - Nalaiya
Database_Users.png (1919×
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint1/output/Database_Users.png
IBM Db2 on Cloud
Data objects
Saved objects
Filter objects
XGF82208
* Untitled - 4
Syntax assistant
Run all
1 select user_id,name,phone_number from users;
History
Results
Result set 1
Details
Filter table
Total:4
USER_ID
NAME
PHONE_NUMBER
16
Dheeraj H
09790571935
11
test_account
76377172773
12
keshav
7387992028
17
charan
987654321
Type here to search
ENG
14:40
16-11-2022


Student Login
IBM
IBM-3989-1662130513
Project Templates - Nalaiya
Database_Skills.png (1919×
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint3/screenshots/Database_Skills.png
IBM Db2 on Cloud
Data objects
Saved objects
Filter objects
XGF82208
* Untitled - 4
Syntax assistant
Run all
1 select * from skills;
History
Results
Result set 1
Details
Filter table
Total:3
SKILL_ID
USER_ID
NAME
44
12
HTML
53
12
CSS
54
12
cloud
Type here to search
ENG
14:47
16-11-2022


Student Login
IBM
IBM-3989-1662130513
Project Templates - Nalaiya
Docker_Image_Registry.png
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint4/screenshots/Docker_Image_Registry.png
docker hub
Search Docker Hub
Explore
Repositories
Organizations
Help
Upgrade
keshav13142
keshav13142
Repositories
ibm_project
Using 0 of 1 private repositories. Get more
General
Tags
Builds
Collaborators
Webhooks
Settings
keshav13142 / ibm_project
Description
Images for react-flask applications.
Last pushed: an hour ago
Docker commands
Public View
To push a new tag to this repository,
docker push keshav13142/ibm_project:tagname
Tags and scans
VULNERABILITY SCANNING - DISABLED
Enable
This repository contains 4 tag(s).
Tag
OS
Type
Pulled
Pushed
v5
Image
an hour ago
an hour ago
v4
Image
an hour ago
2 hours ago
v3
Image
an hour ago
2 hours ago
v1
Image
—
16 days ago
See all
Go to Advanced Image Management
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions.
Upgrade
Learn more
Type here to search
ENG
15:21
16-11-2022


Student Login
IBM
IBM-3989-1662130513
Project Templates - Nalaiya
K8s_Cluster.png (1915×373)
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint4/screenshots/K8s_Cluster.png
Clusters /
ibm
Normal
Expires in 29 days
Add tags
Help
Kubernetes dashboard
Actions...
Overview
Worker nodes
Worker pools
DevOps
New
Pool: Filter...
Search
Add
Name
Status
Worker pool
Zone
Private IP
Public IP
Version
0000007b
Normal
default
Milan 01
10.144.194.182
169.51.207.195
1.24.7_1543
Items per page: 25
1 - 1 of 1 item
1 of 1 page
Type here to search
ENG
15:23
16-11-2022



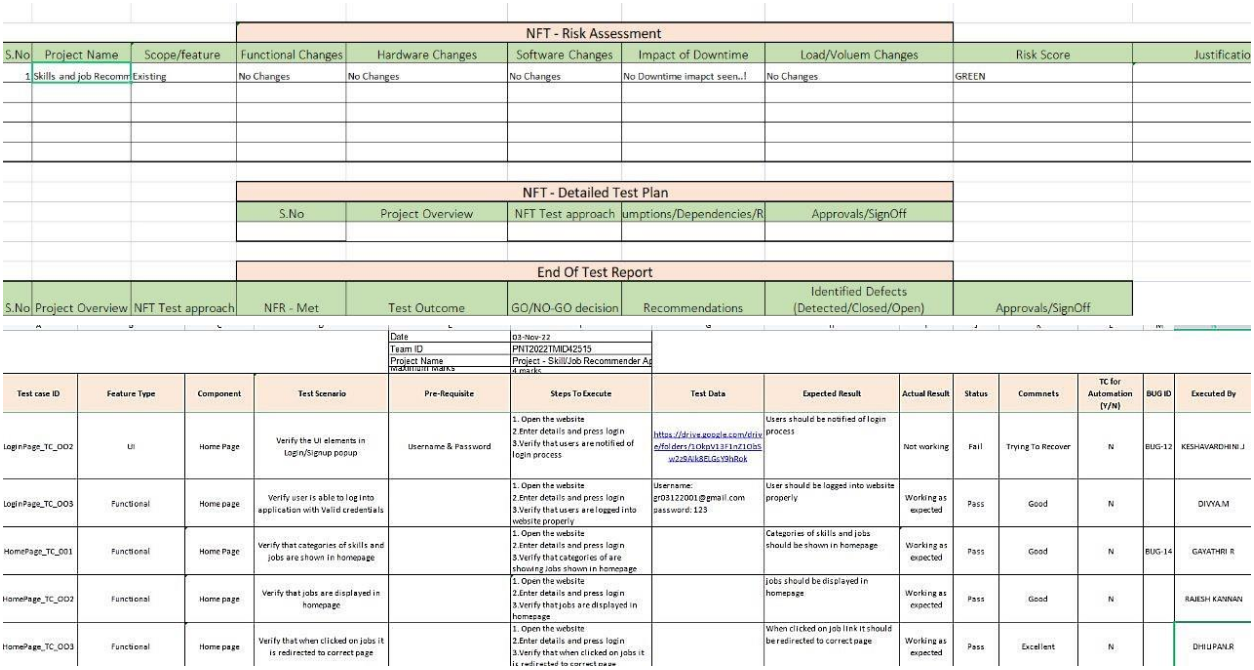


8.TESTING

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.



NFT - Risk Assessment

S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justificatio
1	Skills and job Recomm	Existing	No Changes	No Changes	No Changes	No Downtime imapct seen..!	No Changes	GREEN	

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	umptions/Dependencies/R	Approvals/SignOff

End Of Test Report

S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff

Date	03-Nov-22
Team ID	PNT2022TMID42515
Project Name	Project - Skill/Job Recommender Ap
maximum marks	4 marks

Test case ID	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
LoginPage_TC_OO2	UI	Home Page	Verify the UI elements in Login/Signup popup	Username & Password	1. Open the website 2.Enter details and press login 3.Verify that users are notified of login process	https://drive.google.com/drive/folders/1OkpV13F1nZ1OIaSw2s9Ak8ELGsY9nRok	Users should be notified of login process	Not working	Fail	Trying To Recover	N	BUG-12	KESHAVARDHINI J
LoginPage_TC_OO3	Functional	Home page	Verify user is able to log into application with Valid credentials		1. Open the website 2.Enter details and press login 3.Verify that users are logged into website properly	Username: gr03122001@gmail.com password: 123	User should be logged into website properly	Working as expected	Pass	Good	N		DIVYA.M
HomePage_TC_OO1	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	GAYATHRI R
HomePage_TC_OO2	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		RAJESH KANNAN
HomePage_TC_OO3	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		DHILIPAN.R

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

Acceptance Testing
UAT Execution & Report Submission

Date	13 November 2022
Team ID	PNT2022TMID22285
Project Name	Skills and Job Recommendation
Maximum Marks	4 Marks

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Skills and Job Recommendation project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

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

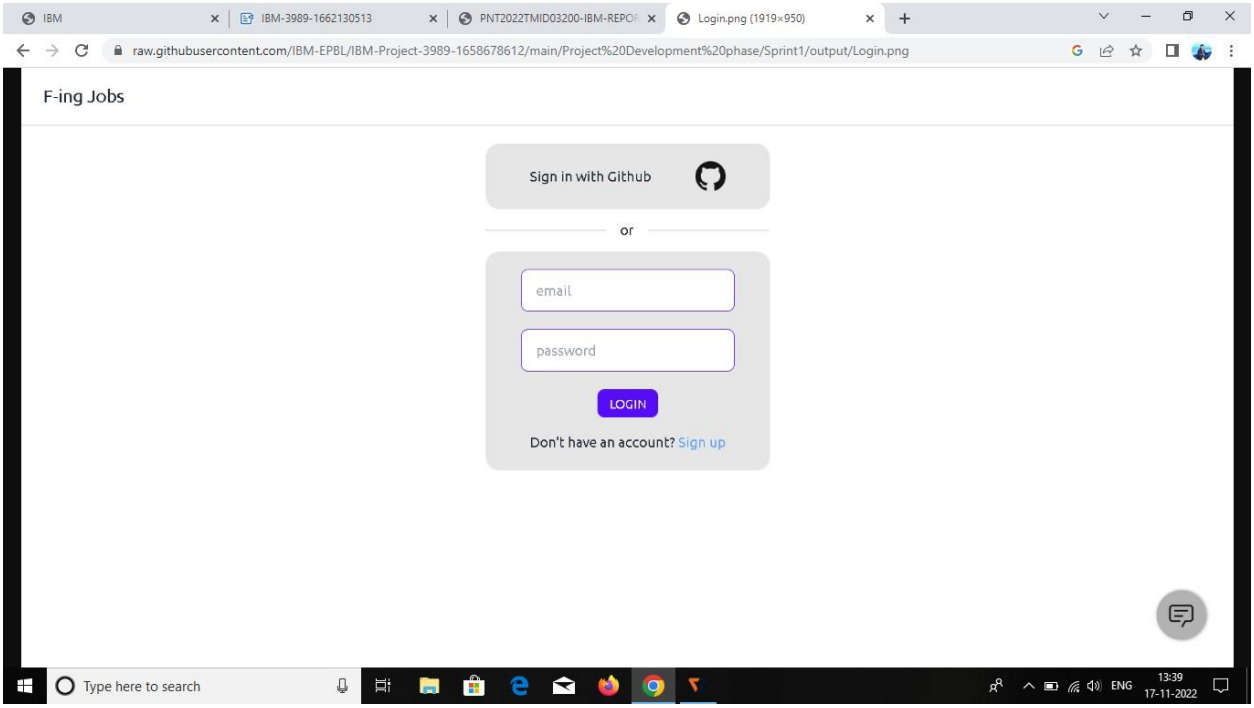
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	1	3	1	6
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	80

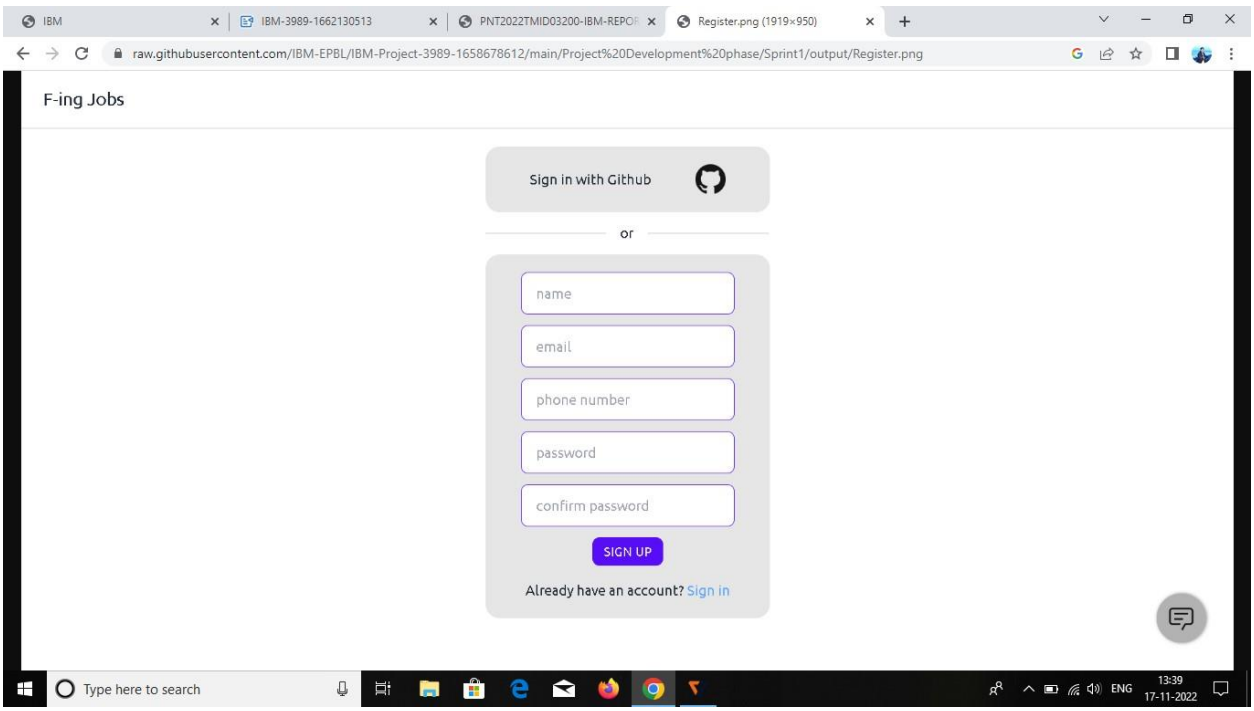
3. Test Case Analysis

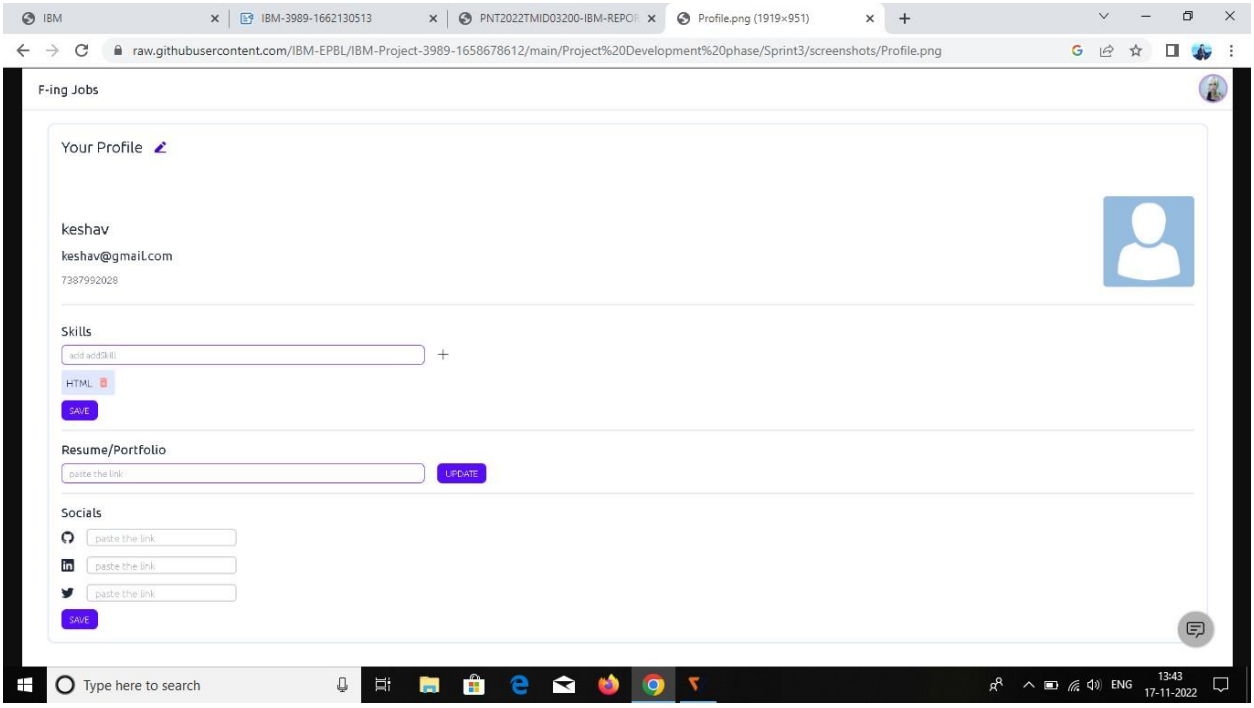
This report shows the number of test cases that have passed, failed, and untested

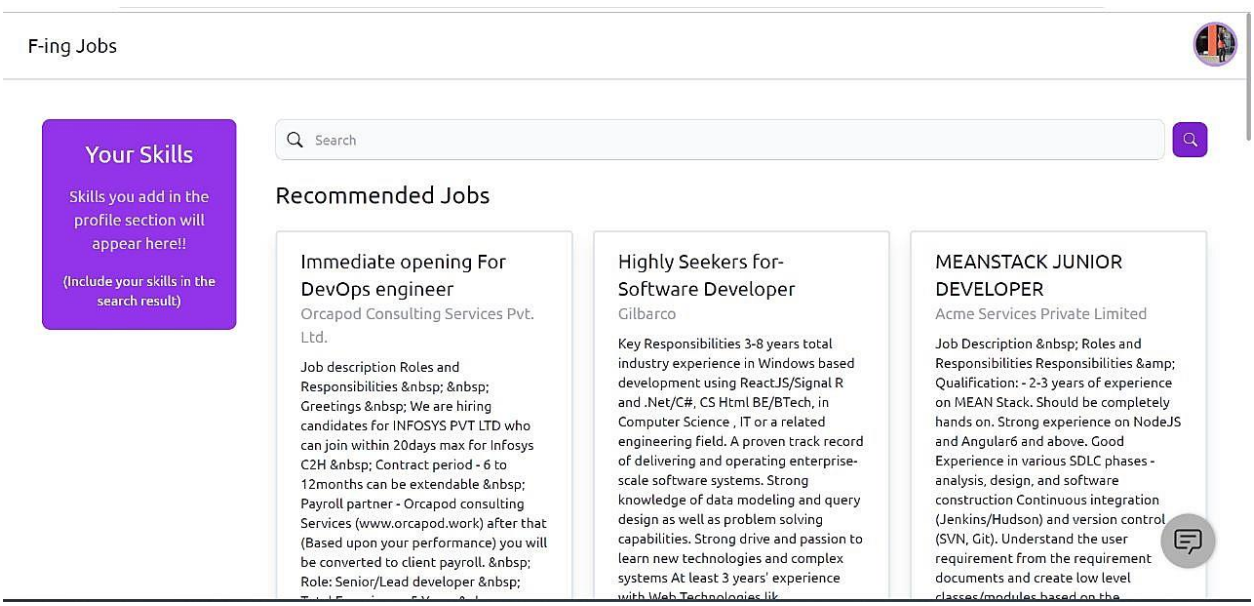
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	1	7
Client Application	51	0	1	51

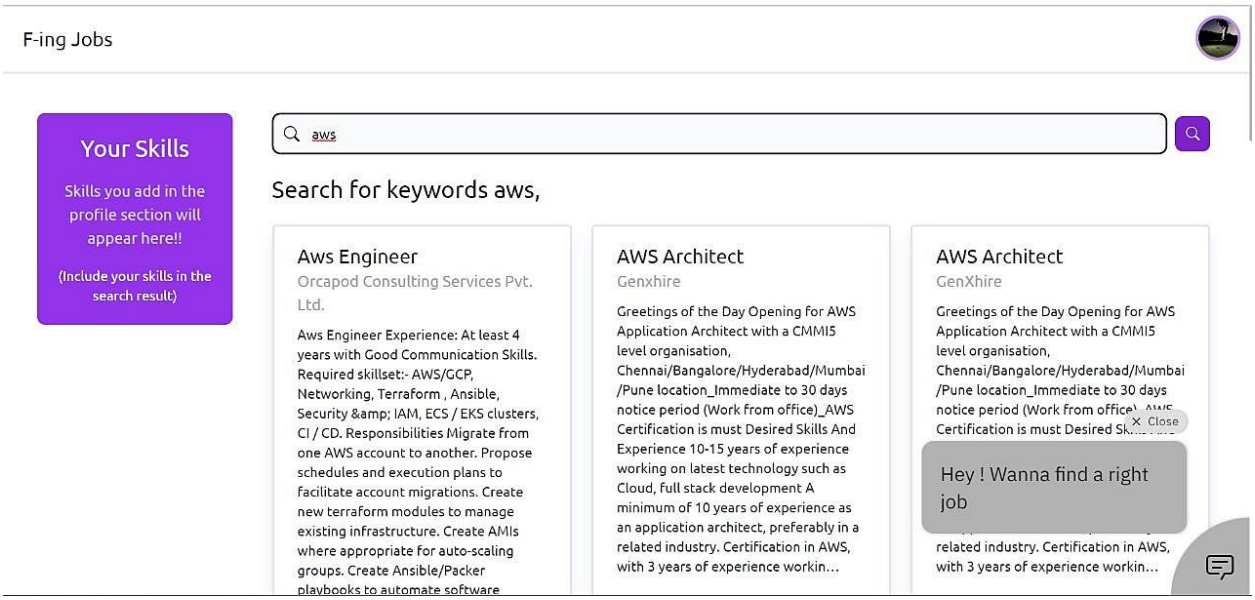
9.RESULTS

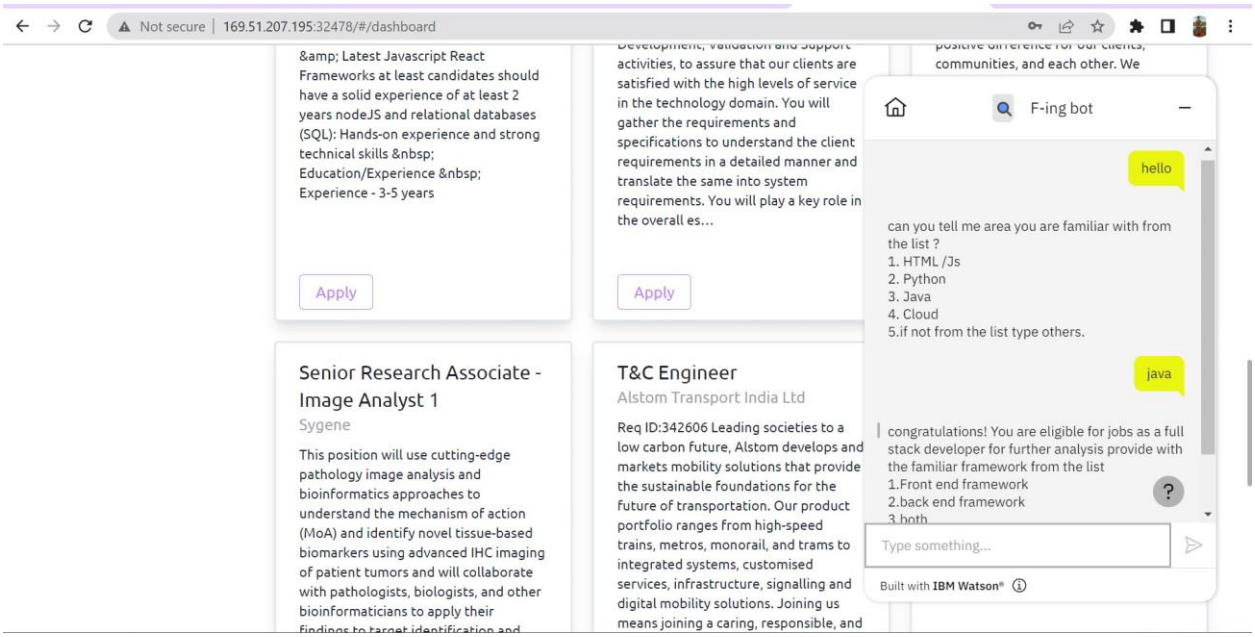

IBM
IBM-3989-1662130513
PNT2022TMID03200-IBM-REPOR
Login.png (1919×950)
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint1/output/Login.png
F-ing Jobs
Sign in with Github
or
email
password
LOGIN
Don't have an account? Sign up
Type here to search
ENG
13:39
17-11-2022

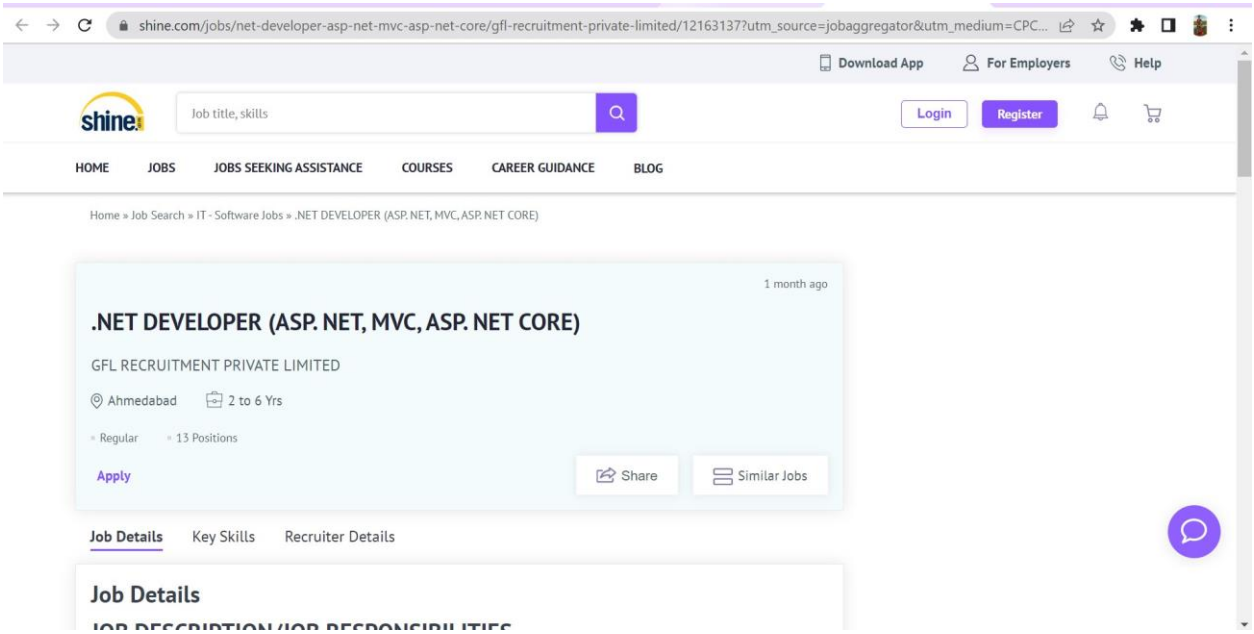

IBM
IBM-3989-1662130513
PNT2022TMID03200-IBM-REPOR
Register.png (1919×950)
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint1/output/Register.png
F-ing Jobs
Sign in with Github
or
name
email
phone number
password
confirm password
SIGN UP
Already have an account? Sign in
Type here to search
ENG
13:39
17-11-2022


IBM
IBM-3989-1662130513
PNT2022TMID03200-IBM-REPO
Profile.png (1919×951)
raw.githubusercontent.com/IBM-EPBL/IBM-Project-3989-1658678612/main/Project%20Development%20phase/Sprint3/screenshots/Profile.png
F-ing Jobs
Your Profile
keshav
keshav@gmail.com
7387992028
Skills
add addSkill
HTML
SAVE
Resume/Portfolio
paste the link
UPDATE
Socials
paste the link
paste the link
paste the link
SAVE
Type here to search
ENG
13:43
17-11-2022


F-ing Jobs
Your Skills
Skills you add in the profile section will appear here!!
(Include your skills in the search result)
Search
Recommended Jobs
Immediate opening For DevOps engineer
Orcapod Consulting Services Pvt. Ltd.
Job description Roles and Responsibilities   Greetings   We are hiring candidates for INFOSYS PVT LTD who can join within 20days max for Infosys C2H   Contract period - 6 to 12months can be extendable   Payroll partner - Orcapod consulting Services (www.orcapod.work) after that (Based upon your performance) you will be converted to client payroll.   Role: Senior/Lead developer
Highly Seekers for-Software Developer
Gilbarco
Key Responsibilities 3-8 years total industry experience in Windows based development using ReactJS/Signal R and .Net/C#, CS Html BE/BTech, in Computer Science , IT or a related engineering field. A proven track record of delivering and operating enterprise-scale software systems. Strong knowledge of data modeling and query design as well as problem solving capabilities. Strong drive and passion to learn new technologies and complex systems At least 3 years' experience
MEANSTACK JUNIOR DEVELOPER
Acme Services Private Limited
Job Description   Roles and Responsibilities Responsibilities & Qualification: - 2-3 years of experience on MEAN Stack. Should be completely hands on. Strong experience on NodeJS and Angular6 and above. Good Experience in various SDLC phases - analysis, design, and software construction Continuous integration (Jenkins/Hudson) and version control (SVN, Git). Understand the user requirement from the requirement documents and create low level


F-ing Jobs
Your Skills
Skills you add in the profile section will appear here!!
(Include your skills in the search result)
aws
Search for keywords aws,
Aws Engineer
Orcapod Consulting Services Pvt. Ltd.
Aws Engineer Experience: At least 4 years with Good Communication Skills. Required skillset:- AWS/GCP, Networking, Terraform , Ansible, Security & IAM, ECS / EKS clusters, CI / CD. Responsibilities Migrate from one AWS account to another. Propose schedules and execution plans to facilitate account migrations. Create new terraform modules to manage existing infrastructure. Create AMIs where appropriate for auto-scaling groups. Create Ansible/Packer playbooks to automate software
AWS Architect
Genxhire
Greetings of the Day Opening for AWS Application Architect with a CMMI5 level organisation, Chennai/Bangalore/Hyderabad/Mumbai /Pune location_Immediate to 30 days notice period (Work from office)_AWS Certification is must Desired Skills And Experience 10-15 years of experience working on latest technology such as Cloud, full stack development A minimum of 10 years of experience as an application architect, preferably in a related industry. Certification in AWS, with 3 years of experience workin...
AWS Architect
GenXhire
Greetings of the Day Opening for AWS Application Architect with a CMMI5 level organisation, Chennai/Bangalore/Hyderabad/Mumbai /Pune location_Immediate to 30 days notice period (Work from office)_AWS Certification is must Desired
Close
Hey ! Wanna find a right job
related industry. Certification in AWS, with 3 years of experience workin...


Not secure | 169.51.207.195:32478/#/dashboard
& Latest Javascript React Frameworks at least candidates should have a solid experience of at least 2 years nodeJS and relational databases (SQL): Hands-on experience and strong technical skills   Education/Experience   Experience - 3-5 years
Apply
activities, to assure that our clients are satisfied with the high levels of service in the technology domain. You will gather the requirements and specifications to understand the client requirements in a detailed manner and translate the same into system requirements. You will play a key role in the overall es...
Apply
communities, and each other. We
F-ing bot
hello
can you tell me area you are familiar with from the list ?
1. HTML /Js
2. Python
3. Java
4. Cloud
5.if not from the list type others.
java
congratulations! You are eligible for jobs as a full stack developer for further analysis provide with the familiar framework from the list
1.Front end framework
2.back end framework
3.both
Type something...
Built with IBM Watson®
Senior Research Associate - Image Analyst 1
Sygene
This position will use cutting-edge pathology image analysis and bioinformatics approaches to understand the mechanism of action (MoA) and identify novel tissue-based biomarkers using advanced IHC imaging of patient tumors and will collaborate with pathologists, biologists, and other bioinformaticians to apply their findings to target identification and
T&C Engineer
Alstom Transport India Ltd
Req ID:342606 Leading societies to a low carbon future, Alstom develops and markets mobility solutions that provide the sustainable foundations for the future of transportation. Our product portfolio ranges from high-speed trains, metros, monorail, and trams to integrated systems, customised services, infrastructure, signalling and digital mobility solutions. Joining us means joining a caring, responsible, and


shine.com/jobs/net-developer-asp-net-mvc-asp-net-core/gfl-recruitment-private-limited/12163137?utm_source=jobaggregator&utm_medium=CPC...
Download App
For Employers
Help
shine
Job title, skills
Login
Register
HOME
JOBS
JOBS SEEKING ASSISTANCE
COURSES
CAREER GUIDANCE
BLOG
Home » Job Search » IT - Software Jobs » .NET DEVELOPER (ASP. NET, MVC, ASP. NET CORE)
1 month ago
.NET DEVELOPER (ASP. NET, MVC, ASP. NET CORE)
GFL RECRUITMENT PRIVATE LIMITED
Ahmedabad
2 to 6 Yrs
Regular
13 Positions
Apply
Share
Similar Jobs
Job Details
Key Skills
Recruiter Details
Job Details

10. ADVANTAGE AND DISADVANTAGE

Advantages:

1.Employment Opportunities:

The foremost advantage of having a profile in our application is that it is your doorway into employment opportunities worldwide. Before the advent of online job applications , students would get jobs through connections. However, now your job opportunities have increased magnanimously. Students who have attained education abroad can put in their area of specialization and find an appropriate job. Apart from this, if there is a particular company that you're interested in, you can make applications for the same.

2.Easy Job Applications:

The traditional recruiting process has taken a back seat and online job application has become paramount. Gone are the days, where you would have to run around with copies of your resume. With the ease of uploading the necessary information on your profile, not only will the recruiters peruse through your profile but you can update your skills regularly. The initial stress of a job application is reduced because the recruiter is already aware of your skills and wants to explore them further. This gives you an excellent opportunity to capitalize on the same and use the app to its fullest.

3.Initiate Connections:

Apart from receiving a job offer, the connections you establish on your profile help you in the long run. For instance, you may start by connecting with your school and college friends and eventually shift to your colleagues. An alumnus from your university is good connections to have. Having an illustrious list of connections speaks to your strong

profile. Having a connection who is working at your dream company can be your pathway to the same. Initiating connections will allow you to analyse industry trends and be at the top of the game.

4.Endorsement and Connections

Collecting endorsements and connections is an excellent way of adding social backing to your profile. As mentioned earlier, having illustrious connections will add value to your profile. Upon receiving endorsement for your skills, employers receive extra confidence in your profile. The trick now is to not only have relevant skills but also make your profile stand out.

Disadvantages:

1. Risk of identity theft

There are loads of personal information that you have to display on your profile for prospective employers to see. Hence, in a case whereby LinkedIn servers develop an issue, you stand a risk of losing important information to the public, resulting in identity theft.

2. Incomplete profile challenge

LinkedIn like other social network websites required you to put up an attractive profile. That is a profile that is appealing to employers and prospective recruiters. People however find it hard to fill out profile details

completely due to one reason or the other.

3. Tons of spam messages

There's a saying that among 12 disciples there will always be a Judas. Think of how many Judas will be available on a website with over 1200 million people. LinkedIn is filled with spam messages from recruiters, employers, and even job seekers. All just to seek attention, mislead, and extort money, etc.

4. Premium package can be expensive

Good thing they say doesn't come cheap. Although, LinkedIn allows you to join the platform without paying. But the LinkedIn premium packages are charged for. For example, the "medium-sized career" price is just about $29.99/month. There are so many added benefits that this offer brings but can still be very costly for a starter or medium-sized business.

11. CONCLUSION

we have used ibm cloud services like db2, cloud registry , kubernetes , Watson assistant to create this application , which will be very usefull for candidates who are searching for job and as well as for the company to select the right candidate for their organization.

12. FUTURE SCOPE

Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation. We can use machine learning technicques to recommend data in a efficient way.

13.APPENDIX

➤ **Source Code:**

index.html:

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

```
<head>
  <meta charset="UTF-8" />
  <link rel="icon" type="image/svg+xml" href="cv.png" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Job Search</title>
</head>

<body>
  <div id="root"></div>
  <script type="module" src="/src/main.jsx"></script>
</body>

</html>
```

Index.css:

```
@import url("https://fonts.googleapis.com/css2?family=Ubuntu&display=swap");

@tailwind base;
@tailwind components;
@tailwind utilities;

:root {
  font-family: Inter, Avenir, Helvetica, Arial, sans-serif;
  font-size: 16px;
  line-height: 24px;
  font-weight: 400;

  color-scheme: light;
  /* color: rgba(255, 255, 255, 0.87);
  background-color: #242424; */

  font-synthesis: none;
  text-rendering: optimizeLegibility;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  -webkit-text-size-adjust: 100%;
}
* {
```

```
  margin: 0;
  padding: 0;
  font-family: "Ubuntu", sans-serif;
}

body::-webkit-scrollbar {   width:
5px;
  background-color: none;
  border-radius: 20px;
}

body::-webkit-scrollbar-thumb {   background-color:
#adadad;
  border-radius: 20px;
}

body {
  max-height: 100vh;
}
```

Main.py:

```
from backend import create_app import os

app = create_app()

port = os.environ.get("PORT", 5000)


if __name__ == '__main__':
    from waitress import serve
    serve(app, port=port)
```

App.jsx:

```
import { useEffect } from "react";
import { HashRouter, Route, Routes } from "react-router-dom";
import Navbar from "./components/Navbar"; import {
AppProvider } from "./context/AppContext"; import
```

```
Auth from "./screens/Auth"; import Dashboard from
"./screens/Dashboard"; import Profile from
"./screens/Profile";


function App() {   useEffect(()
=> {
   window.watsonAssistantChatOptions = {
     integrationID: import.meta.env.VITE_WATSON_INTEGRATION_ID, // The ID of this integration.
region: import.meta.env.VITE_WATSON_REGION, // The region your integration is hosted in.
serviceInstanceID: import.meta.env.VITE_WATSON_SERVICE_INSTANCE_ID, // 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);
   });
 }, []);
 return (
   <HashRouter>
     <AppProvider>
       <Navbar />
       <Routes>
         <Route path="/" element={<Auth />} />
         <Route path="/dashboard" element={<Dashboard />} />
         <Route path="/profile" element={<Profile />} />
       </Routes>
     </AppProvider>
   </HashRouter>
 );
}
```

```
export default App;
```

➤ **CONTEXT**

appContent.jsx

```
import { createContext, useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";

export const AppContext = createContext();

export const AppProvider = ({ children }) => {
 const navigate = useNavigate();

 const [skills, setSkills] = useState([]);

 const [user, setUser] = useState(null);

 useEffect(() => {
  let temp_user = JSON.parse(localStorage.getItem("user"));
  if (!temp_user) {
   navigate("/");
  } else {
   setUser(temp_user);
  }
 }, []);
 return (
  <AppContext.Provider value={{ user, setUser, skills, setSkills }}>
   {children}
  </AppContext.Provider>
 );
};
```

➤ **proxies**

backendapi.js

```
import { BASE_URL } from "../utils/helper";

export const loginUser = async (inputs) => {
  try {
    const response = await fetch(`${BASE_URL}/auth/login`, {
      method: "POST",
      body: JSON.stringify(inputs),
      headers: {
        "Content-Type": "application/json",
      },
    });
    const data = await response.json();
    return data;
  } catch (error) {
    console.error(error);
  }
};

export const registerUser = async (inputs) => {
  try {
    const response = await fetch(`${BASE_URL}/auth/signup`, {
      method: "POST",
      body: JSON.stringify(inputs),
      headers: {
        "Content-Type": "application/json",
      },
    });
    const data = await response.json();
    return data;
  } catch (error) {
    console.error(error);
```

```
  }
};

export const getUserSkills = async (token) => {
  try {
    const response = await fetch(`${BASE_URL}/user/skills`, {
      method: "GET",
      headers: {
        Authorization: `Bearer ${token}`,
        "Content-Type": "application/json",
      },
    });
    if (response.ok) {
      const { skills } = await response.json();
      return skills;
    } else {
      return null;
    }
  } catch (error) {
    console.error(error);
  }
};

export const saveUserSkills = async (skills, token) => {
  try {
    const response = await fetch(`${BASE_URL}/user/skills`, {
      method: "POST",
      body: JSON.stringify({ skills }),
      headers: {
        Authorization: `Bearer ${token}`,
        "Content-Type": "application/json",
      },
    });
    if (response.ok) {
```

```
      return true;
    } else {
      return false;
    }
  } catch (error) {
    console.error(error);
  }
};

export const removeUserSkills = async (skills, token) => {
  try {
    const response = await fetch(`${BASE_URL}/user/skills`, {
      method: "DELETE",
      body: JSON.stringify({ skills }),
      headers: {
        Authorization: `Bearer ${token}`,
        "Content-Type": "application/json",
      },
    });
    if (response.ok) {
      return true;
    } else {
      return false;
    }
  } catch (error) {
    console.error(error);
  }
};

export const updateUserDetails = async (inputs, token) => {
  try {
    const response = await fetch(`${BASE_URL}/user/profile`, {
      method: "POST",
      body: JSON.stringify(inputs),
```

```
    headers: {
      Authorization: `Bearer ${token}`,
      "Content-Type": "application/json",
    },
   });
   if (response.ok) {
    const data = await response.json();
    return data;
   } else {
    return null;
   }
  } catch (error) {
   console.error(error);
  } };
```

➤ **Docker file**

```
# Build step #1: build the React front end
FROM node:16-alpine as react-builder
WORKDIR /app
ENV PATH /app/node_modules/.bin:$PATH
COPY package.json ./
COPY ./src ./src
COPY ./public ./public
COPY ./index.html ./vite.config.js ./postcss.config.cjs ./tailwind.config.cjs ./.env  ./
RUN npm install
RUN npm run build

# Build step #2: build the API with the client as static files
FROM python:3.10
WORKDIR /app
COPY --from=react-builder /app/dist ./dist
COPY main.py ./main.py

RUN mkdir ./backend
```

```
COPY backend/ ./backend/
RUN pip install -r ./backend/requirements.txt

EXPOSE 5000
ENTRYPOINT ["python","main.py"]
```

➤ **main.py**

```
from backend import create_app import os

app = create_app()

port = os.environ.get("PORT", 5000)


if __name__ == '__main__':
   from waitress import serve
   serve(app, port=port)
```

- **OUTPUT LINK:** **http://169.51.207.195:32478/**

➤ **GITHUB Link :** https://github.com/IBM-EPBL/IBM-Project-21451-1659780497

☞ **Project Demo link:**
https://drive.google.com/file/d/1vbgB7azE6w2L_WGZHgua9iMphuuYyaYH/view?usp=share_link

THANK YOU