SKILL / JOB RECOMMENDER APPLICATION

PROJECT REPORT

Submitted by

CHANDRU K	**-**	**201912001**
GIRIHARAN K	**-**	**201912005**
KESAVARAMAN K	**-**	**201912011**
RISHIKUMAR K	**-**	**201912019**
SUJI D	**-**	**201912026**

in partial fulfillment of the requirements for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

SENGUNTHAR ENGINEERING COLLEGE

TIRUCHENGODE

(An Autonomous Institution)

ANNA UNIVERSITY, CHENNAI NOV 2022

Skill /Job Recommender Application

1. INTRODUCTION

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.

1.1 Project Overview

There has been a sudden boom in the technical industry and an increase in the number of good start-ups. Keeping track of various appropriate job openings in top industry names has become increasingly troublesome. This leads to deadlines and hence important opportunities being missed. Through this research paper, the aim is to automate this process to eliminate this problem. To achieve this, IBM cloud services like db2, Watson assistant, cluster, Kubernetes have been used. A hybrid system of Content-Based Filtering and Collaborative Filtering is implemented to recommend these jobs. The intention is to aggregate and recommend appropriate jobs to job seekers, especially in the engineering domain. The entire process of accessing numerous company websites hoping to find a relevant job opening listed on their career portals is simplified. The proposed recommendation system is tested on an array of test cases with a fully functioning user interface in the form of a web application. It has shown satisfactory results, outperforming the existing systems. It thus testifies to the agenda of quality over quantity.

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 anabundance of choices and not much streamlining. On the basis of the actual skills or interests of an individual, job seekers often find themselves unable tofind the appropriate employment for themselves. This system, therefore, approaches the idea from a data point of view, emphasizing more on the qualityof the data than the quantity.

2. LITERATURE SURVEY

2.1 Existing Problem

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.

[1] In this section, we describe our framework for job recommendation. We narrow down the scope and focus on recommendation of job vacancies for Information Technology (IT) professionals acting in the Brazilian market. The proposed framework is composed by three stages: data collection, data preparation and recommendation.

[2] we select a group of the nearest job offers based on the distance to that profile (job matching). In the case of TF-IDF representation, we use the cosine distance while for word embeddings, we use the relatively new Word Mover’s Distance (WMD) [Kus15]. Once retrieved the top ”k” job offers for the profile, we sort them in descending order based on the inverse of this distance (ranking)

[3] To perform job offers scraping, we created a list of keywords from the IT industry and used them as search terms. For each keyword, we search all the

related job offers using Catho's search engine and save the retrieved results in our database; thus, the content's quality is highly related to the quality of the Catho's search engine.

[4] we retrieved data from job search sites using only IT keywords, there were still some job offers that do not correspond to this field, then, the first step in this phase is filtering out job offers that do not belong to the IT field. To achieve this, we use a dictionary of weighted IT terms to match each job offer in its document-like format.

[5] Once job offers and profiles are filtered, the second step is text pre-processing. In this task, we perform stop words removal, tokenization and lemmatization for the Portuguese language.

[6] The third step, feature representation, aims to represent these documents (job offers and profiles) as vector space models. For this purpose, we adopted two approaches: word embeddings and TF-IDF. The latter technique does not require so much effort to be implemented unlike the former.

2.2 References

Shaha 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 PROBLEM

2.3 Problem Statement Definition

"Can an efficient recommender system be modeled for the Job seekers which recommend Jobs with the user’s skill set and job domain and also addresses the issue of cold start?".

In current situation recruitments done manually for lakhs of students in which many talented students may lose their opportunities due to different reasons since it is done manually, and company also need the highly talented people from the mass group for their growth. So we have built a cloud application to do this process in a efficient manner.

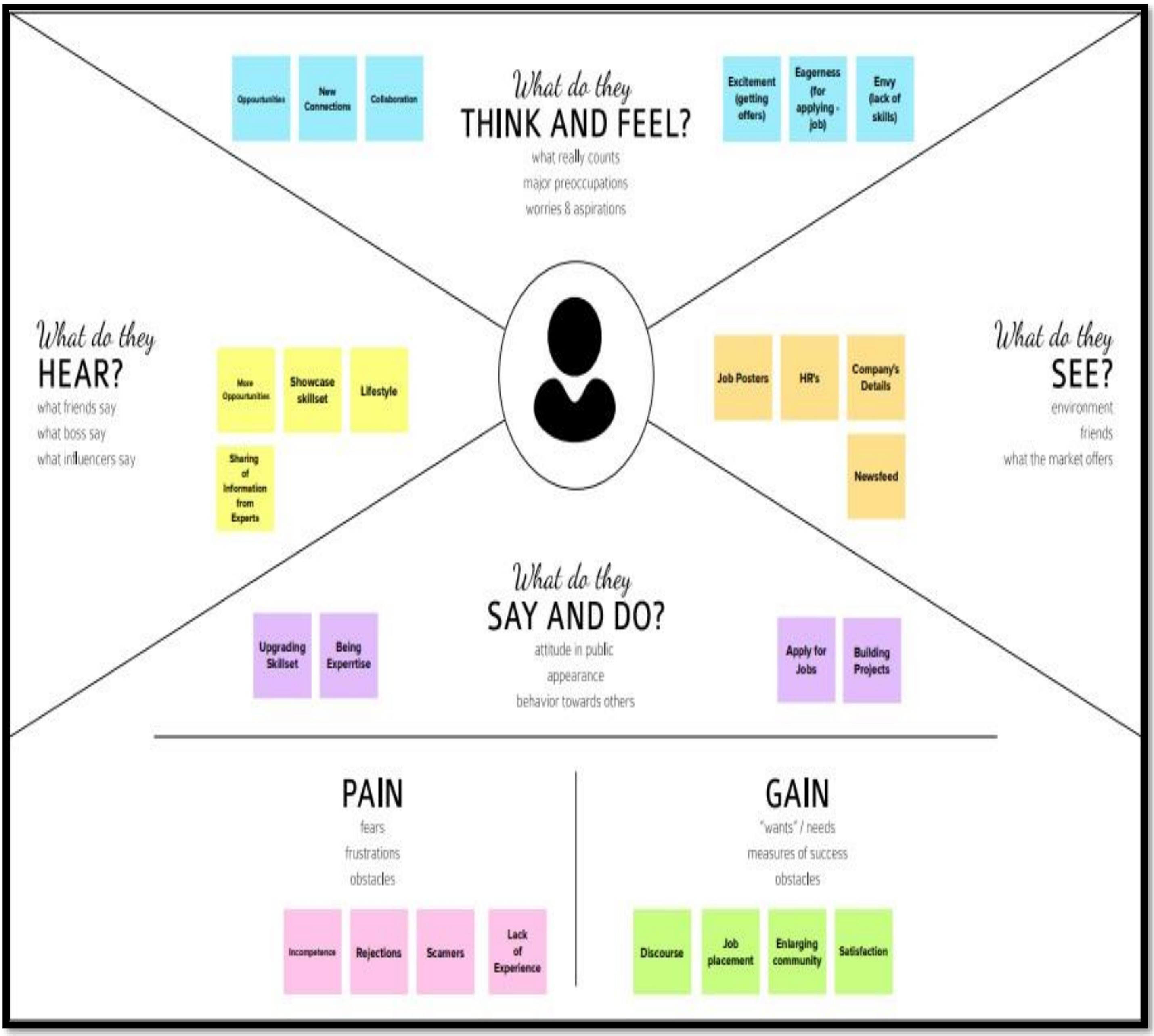
3. IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a collaborative visualization used to articulate what we know about a particular type of user. It externalizes knowledge about users in order to

1) Create a shared understanding of user needs, and

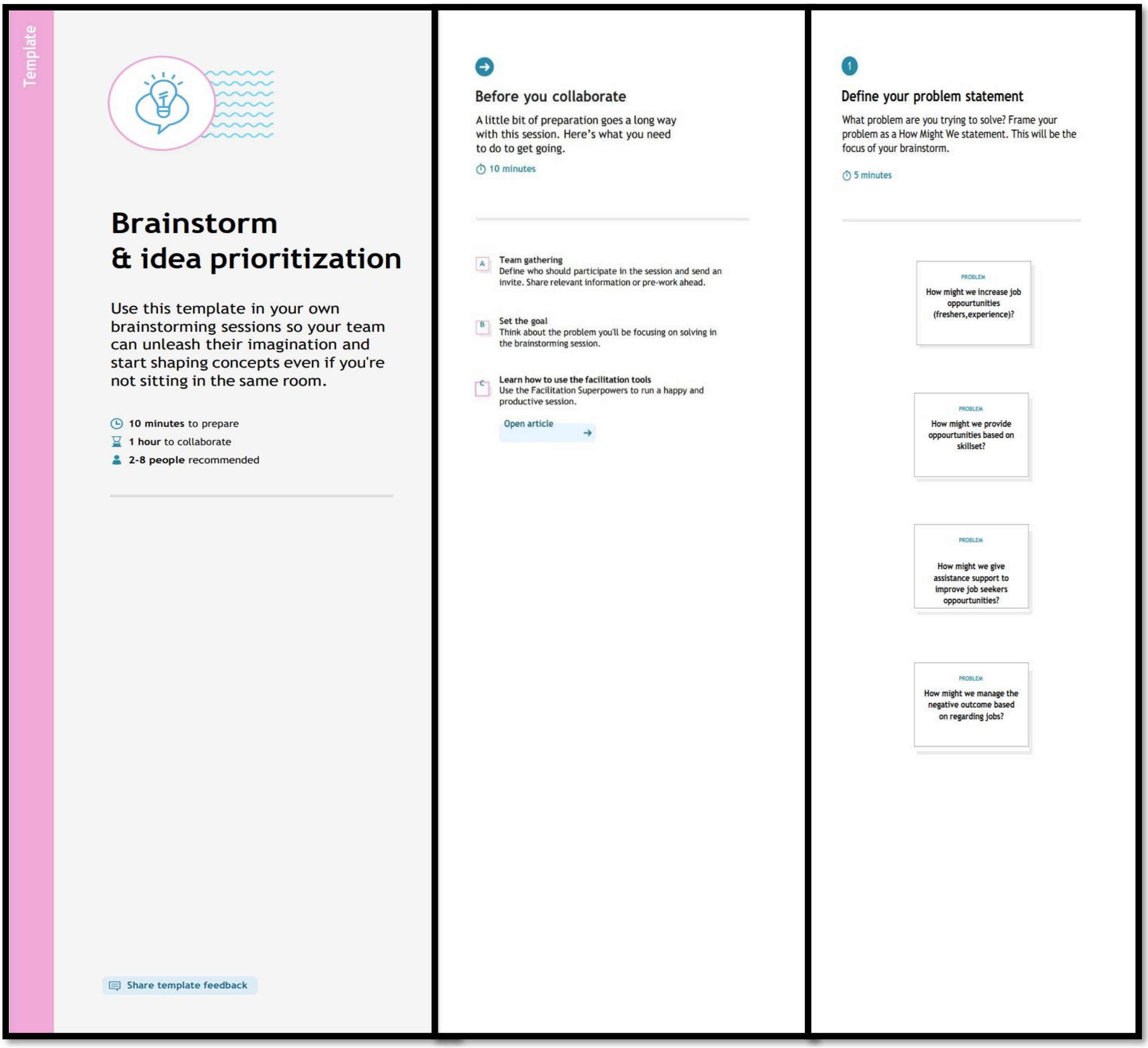
2) Aid in decision making



3.2 Ideation & Brainstorming

Brainstorm & Idea Prioritization Template: Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. 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.

Step 1: Team Gathering, Collaboration and Select the Problem Statement



Brainstorm & idea prioritization

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

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

Before you collaborate

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

10 minutes

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

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

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

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

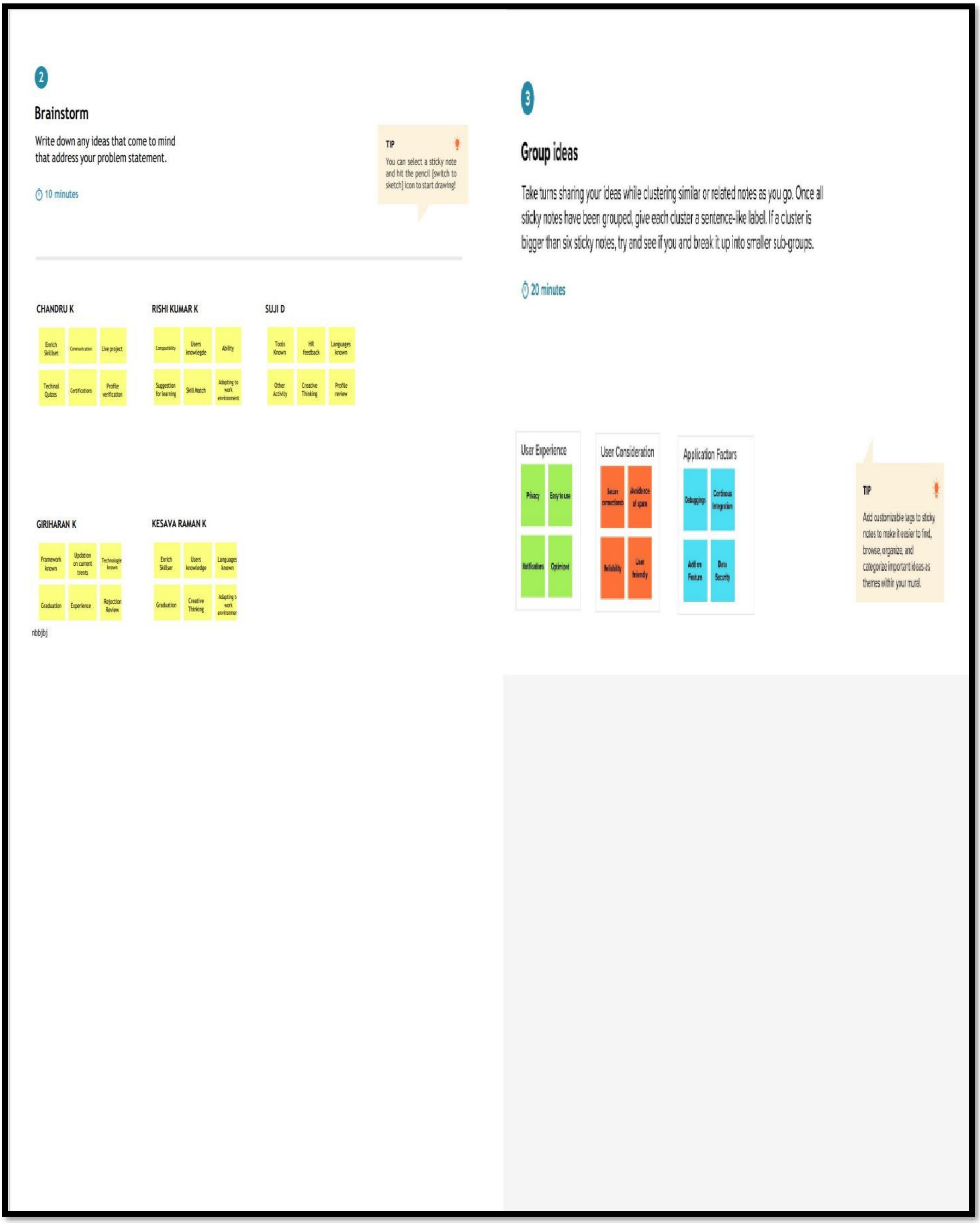
PROBLEM
How might we increase job oppourtunities (freshers,experience)?

PROBLEM
How might we provide oppourtunities based on skillset?

PROBLEM
How might we give assistance support to improve job seekers oppourtunities?

PROBLEM
How might we manage the negative outcome based on regarding jobs?

Step 2: Brainstorm, Idea Listing and Grouping



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!

CHANDRU K

Enrich Skillset | Communication | Live project

Techinal Quizes | Certifications | Profile verification

RISHI KUMAR K

Compatibility | Users knowlegde | Ability

Suggestion for learning | Skill Match | Adapting to work environment

SUJI D

Tools Known | HR feedback | Languages known

Other Activity | Creative Thinking | Profile review

GIRIHARAN K

Framework known | Updation on current trents | Technologie known

Graduation | Experience | Rejection Review

KESAVA RAMAN K

Enrich Skillser | Users knowledge | Languages known

Graduation | Creative Thinking | Adapting t work environmer

nbbjbj

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, 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

User Experience

Privacy | Easy to use

Notifications | Optimized

User Consideration

Secure connections | Avoidence of spam

Reliability | User friendly

Application Factors

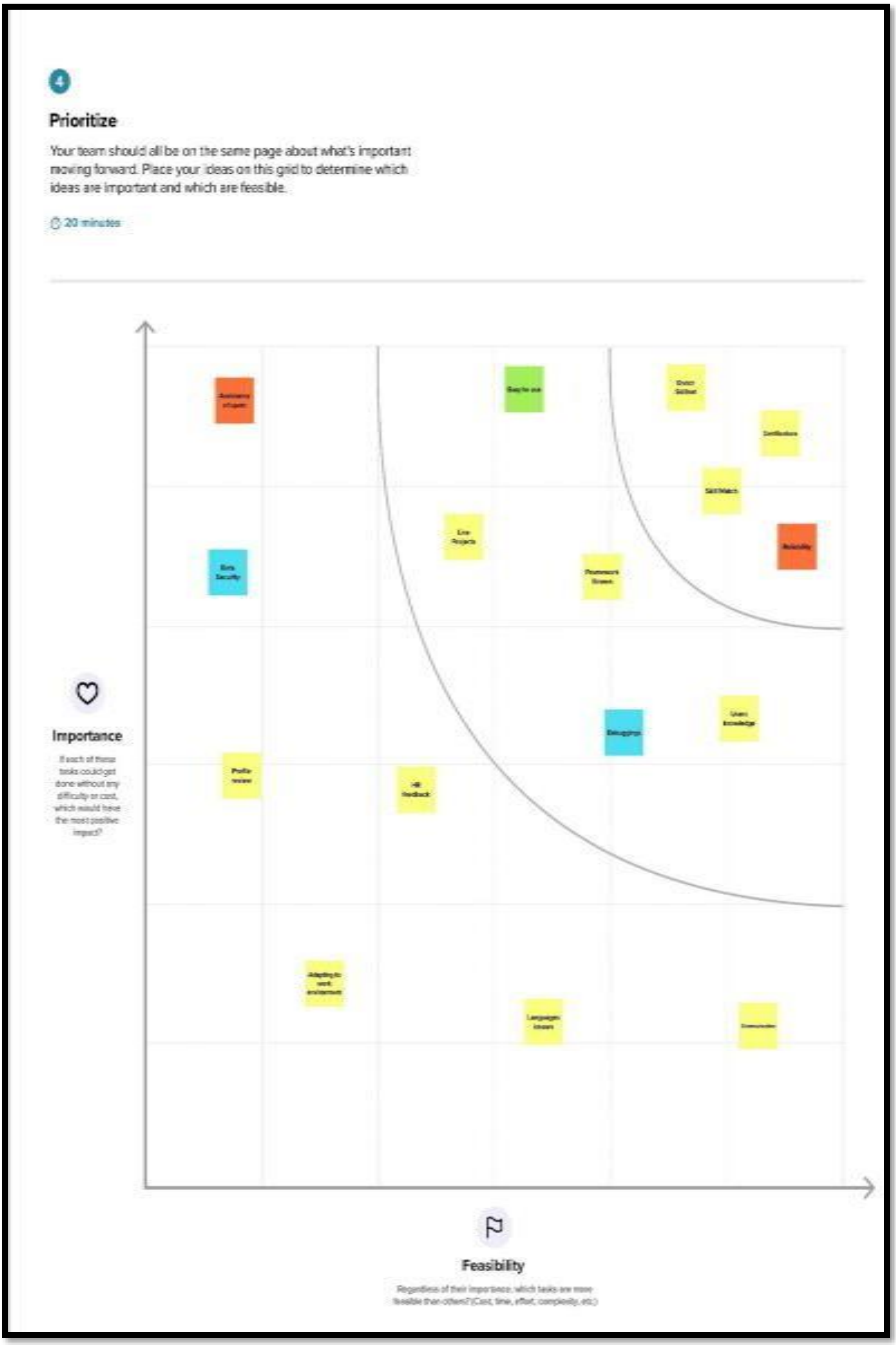
Debuggings | Continous Integration

Add on Feature | Data Security

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Step 3: Idea Prioritization



4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance

If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility

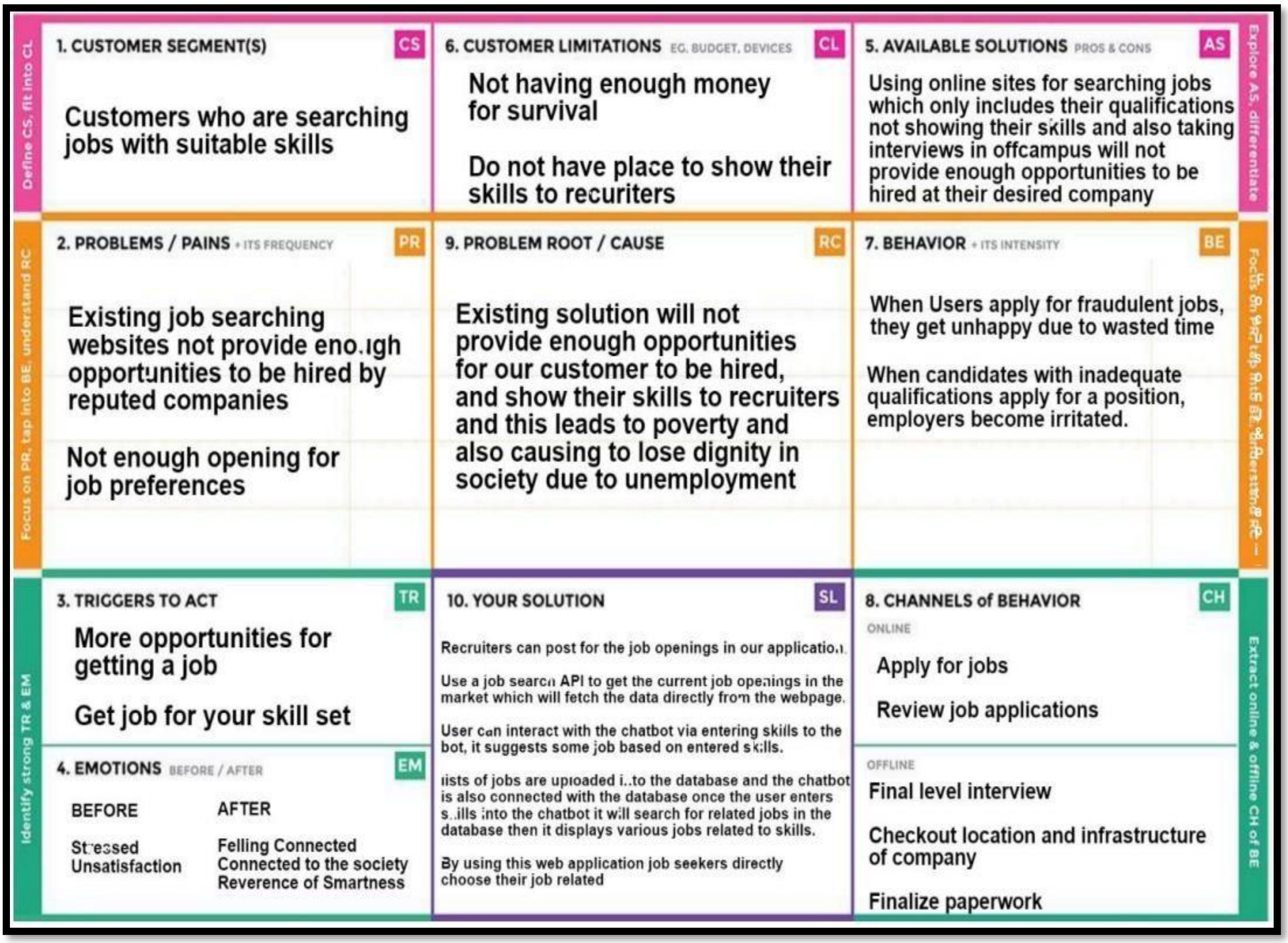
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

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

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

3.4 Problem Solution Fit



Define CS, fit into CL	1. CUSTOMER SEGMENT(S) CS Customers who are searching jobs with suitable skills	6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL Not having enough money for survival Do not have place to show their skills to recuriters	5. AVAILABLE SOLUTIONS PROS & CONS AS Using online sites for searching jobs which only includes their qualifications not showing their skills and also taking interviews in offcampus will not provide enough opportunities to be hired at their desired company	Explore AS, differentiate
Focus on PR, tap into BE, understand RC	2. PROBLEMS / PAINS + ITS FREQUENCY PR Existing job searching websites not provide eno.ıgh opportunities to be hired by reputed companies Not enough opening for job preferences	9. PROBLEM ROOT / CAUSE RC Existing solution will not provide enough opportunities for our customer to be hired, and show their skills to recruiters and this leads to poverty and also causing to lose dignity in society due to unemployment	7. BEHAVIOR + ITS INTENSITY BE When Users apply for fraudulent jobs, they get unhappy due to wasted time When candidates with inadequate qualifications apply for a position, employers become irritated.	Focus on PR, tap into BE, understand RC
Identify strong TR & EM	3. TRIGGERS TO ACT TR More opportunities for getting a job Get job for your skill set 4. EMOTIONS BEFORE / AFTER EM BEFORE: Stressed, Unsatisfaction AFTER: Felling Connected, Connected to the society, Reverence of Smartness	10. YOUR SOLUTION SL Recruiters can post for the job openings in our applicatio.ı Use a job searcıı API to get the current job openings in the market which will fetch the data directly from the webpage. User can interact with the chatbot via entering skills to the bot, it suggests some job based on entered skills. ıists of jobs are upıoaded i..to the database and the chatbot is also connected with the database once the user enters s..ills into the chatbot it will search for related jobs in the database then it displays various jobs related to skills. By using this web application job seekers directly choose their job related	8. CHANNELS of BEHAVIOR CH ONLINE Apply for jobs Review job applications OFFLINE Final level interview Checkout location and infrastructure of company Finalize paperwork	Extract online & offline CH of BE

4.REQUIREMENT ANALYSIS

4.1 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 through Form. Registration through Gmail. Registration throughApplication.
FR-2	User Confirmation	Confirmation via Email. Confirmation via OTP.
FR-3	Newsfeed	Uploading title,description,Images through form.
FR-4	Skill Match	Checks for the skill listed in the Job opening and send an notification to user.
FR-5	Job Search	Takes input from the form and filter with the openings in the Database and list it.
FR-6	Chatbot	To Solve user queries and problems and helps in job search.

4.2 Non-functional requirements:

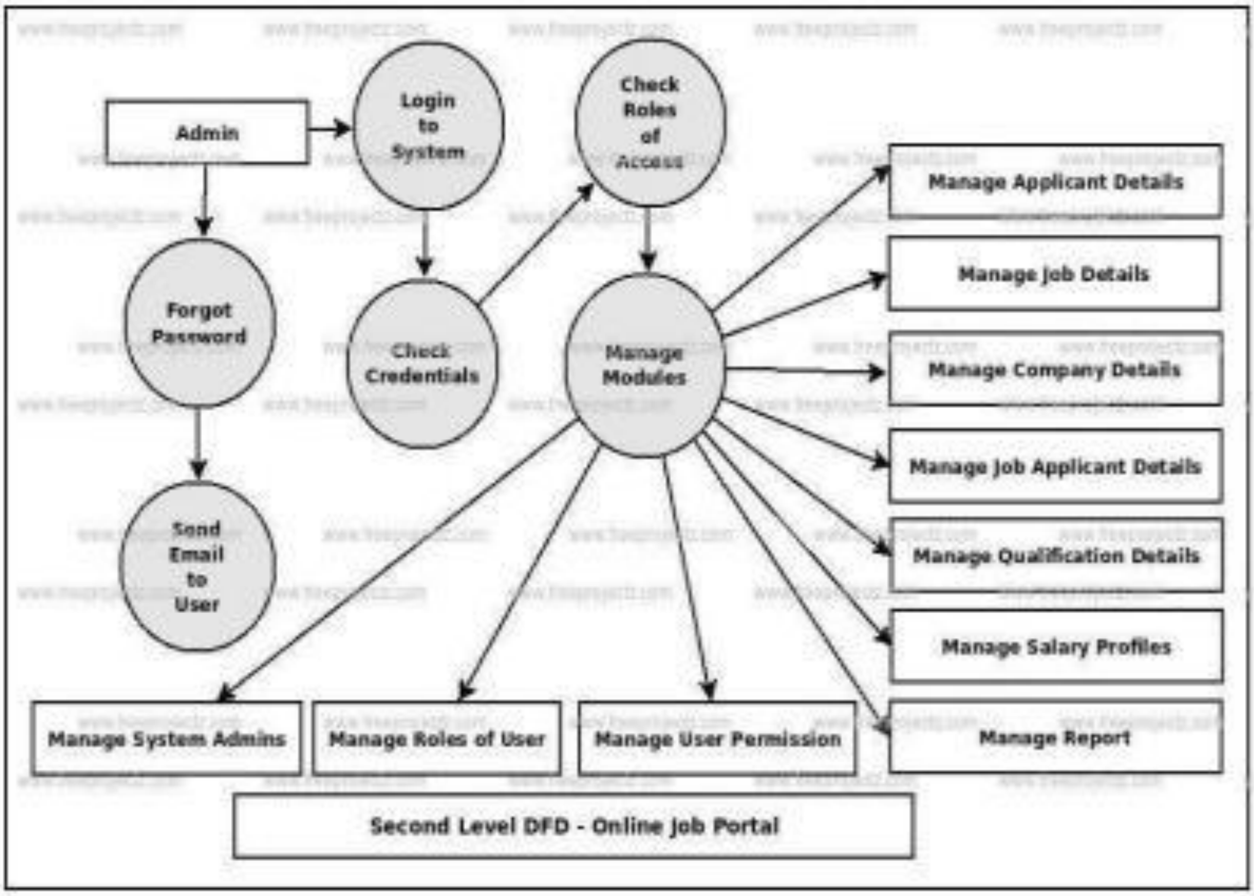
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	This application can be used by the job seekers to search for the job based on their skillset and recuriters can post their job openings.
NFR-2	Security	Every Routes will be protected i.e an Seeker cannot use the functionality of Recuriterrs and vice versa. passwords are hashed
NFR-3	Reliability	Large number of job opening will be provided with trusted end to end connection to all the job seekers.
NFR-4	Performance	Optimized based on time and space complexity
NFR-5	Availability	For job seekers who has passion to their skillset.
NFR-6	Scalability	Scaled using microservice architecture.

5.PROJECT DESIGN

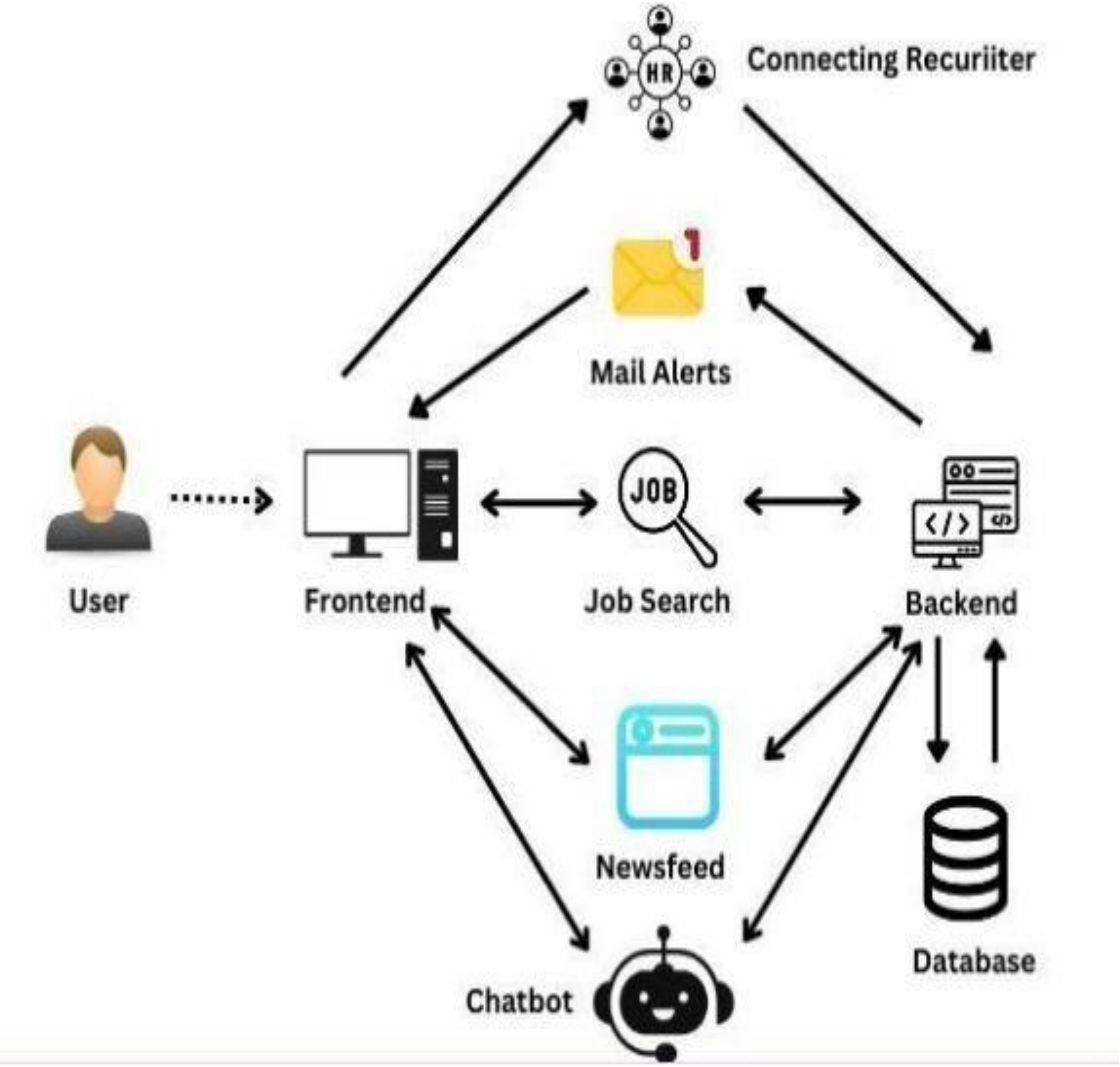
5.1 Data flow diagrams:

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



5.2 Solution Technology Architecture:

The deliverable shall include the architectural diagram as below



S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js /ReactJ setc.
2.	Developing Interface	Developing application for the task	Java /Python
3.	Voice Assistance	Voice commands instead of typing.	IBM Watson STT service
4.	Chatbot Assistance	Conversational Interface	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBMDB2, IBM Cloudant etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local File system
8.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
9.	Infrastructure (Server/Cloud)	Application Deployment on Local System /Cloud Local Server Configuration: Cloud Server Configuration:	Local, Cloud Foundry, Kubernetes, etc.

5.3 User Stories

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

User Type	Functional Requirement	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 & Password	I can receive confirmation email & click confirm	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	Genera Dashboard for job seekers through this dashboard see the jobs applied and the status of jobs.	Job seekers: I can see the job status.	High	Sprint-1
		USN-7	The UI which more flexible in large and small devices	Access through all devices	High	Sprint-1
	Search or Filter	USN-8	The filter section gives more flexibility to a searching job and time saver.	As a user, I can search for the desired company's.	Medium	Sprint-2
	Apply	USN-9	As a user, I can apply for a company and submit the application.	Able to ably jobs.	High	Sprint-2

Customer Care Executive	Bot or Virtual Assistant	USN-10	As an executive to provide a quality-based service chatbot is important for assisting if any assistance is needed for the user	Able to handle user queries	Medium	Sprint-3
Administrator	Admin Dashboard	USN-11	Admin can make change the job posts and able to update the job is active or inactive.	Able to update, delete and change the status of the job posts.	High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING

Planning - Planning pertains to the process of creating a plan of which materials and resources will be required to fulfil incoming and forecasted demand. This step is crucial to ensure that you have enough materials and resource capacity available to produce your orders on time. This component pertains to the 'what' and 'how' of any project: what exactly needs to be achieved and how it will be accomplished.

Scheduling - Scheduling pertains to establishing the timing of the use of specific resources of that organization. In production, scheduling involves developing schedules for workers, equipment, and materials. It reflects on the 'when' of a 29 project, by assigning the appropriate resources to get the production plan completed within a period of time. Creating optimized production schedules ensures that your facility is able to reduce costs, increase productivity, and deliver goods to customers on time.

In order to create accurate and realistic production plans that allow manufacturers to react quickly to changes, it is important to have a production plan that is aligned with the resource and material scheduling process. Having any discrepancy or divergence between the planning and scheduling process creates inefficiencies that can be costly for your business. The bigger the divergence, the larger the cost.

6.1 Sprint Planning and Estimation

Planning:

In Sprint Planning, the team decides what it will build in the upcoming Sprint and how they will build it. The team commits to the Sprint goal after breaking down user stories into tasks and doing task-level estimation. Sprint Planning is done by the Product Owner, Scrum Master, and the Team. In Scrum, every project is broken into time blocks called sprints, usually 2-4 weeks long. A sprint planning meeting is when the team (including the Scrum Master, Scrum Product Manager, and Scrum Team) meets to determine which backlog items will be handled in the next sprint.

Estimation:

In Scrum Projects, Estimation is done by the entire team during Sprint Planning Meeting. The objective of the Estimation would be to consider the User Stories for the Sprint by Priority and by the Ability of the team to deliver during the Time Box of the Sprint.

Product Owner ensures that the prioritized User Stories are clear, can be subjected to estimation, and they are brought to the beginning of the Product Backlog.

As the Scrum Team in total is responsible for the delivery of the product increment, care would be taken to select the User Stories for the Sprint based on the size of the Product Increment and the effort required for the same. The size of the Product Increment is estimated in terms of User Story Points. Once the size is determined, the effort is estimated by means of the past data, i.e., effort per User Story Point called Productivity.

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	**UI / UX Design**	**USN-1**	**As a user, I need to interactwith websites. So, this willbe the interface between me and website.**	**High**	**I can interact withwebsite**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman.**
Sprint-1	**Registration**	**USN-2**	**As a user, I can register forthe application by entering my email, password, and confirming my password.**	**High**	**I can register withemail**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman.**
Sprint-1		**USN-3**	**As a user, I will receive confirmation email once Ihave registered for the application**	**High**	**I can receive confirmation mail**	**Chandru, Rishikumar, Suji, Giriharan , Kesavaraman**
Sprint-1	**Login**	**USN-4**	**As a user, I can log into theapplication by entering email & password**	**Low**	**I can login to the application by entering login credentials**	**Chandru, Rishikumar , Suji, Giriharan, Kesavaraman**
Sprint-1	**Flask**	**USN-5**	**As a user, I can access thewebsite in quick span of time.**	**High**	**I can access website quickly**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**

Sprint-1	**Dashboard**	**USN-6**	**As a user, I can access various tools and serviceson the website through the help of dashboard.**	**High**	**I can access various tools andservices**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
			Submission of sprint 1			
Sprint-2	**Database**	**USN-7**	**As a user, I can store thedata am providing in theportal.**	**High**	**I can store my data**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-2	**User Profile**	**USN-8**	**As a user, I can update myprofile in the user**	**Medium**	**I can modify mydetails**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
			profile section			
Sprint-2	**SendGrid Integration**	**USN-9**	**As a user, I can get notification through mailservices**	**Medium**	**I can get mail**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-2	**Learning Resources**	**USN-10**	**As a user, I can access resources and knowledge which is useful in developing the skills**	**Medium**	**I can access learning resources**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-2	**Link to pythoncode**	**USN-11**	**As a user, I can access website fast**	**High**	**I can access website**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
			Submission of sprint 2			

Sprint-3	**Cloud Storage**	**USN-12**	**As a user, I can store my photos, resumes and much more media that are supported in the webpage.**	**High**	**I can store the medias**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-3	**Chat-Bot**	**USN-13**	**As a user, I can access thechatbot which is very much useful in knowing several things like recent available jobs and jobs opening etc..**	**High**	**I can interact withchatbot to solve any queries**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-3	**Integrate to App**	**USN-14**	**As a user, I can interact with chatbot after the integration of chat-bot and webpage.**	**High**	**Easy access withchat-bot**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
			Submission of Sprint 3			
Sprint-4	**Docker**	**USN-15**	**As a user, I can access thewebpage in any device.**	**High**	**I can access myaccount in any device**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-4	**Kubernetes**	**USN-16**	**As a user, I can access thewebpage in any device.**	**High**	**I can access my account in anydevice**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-4	**Deployment incloud**	**USN-17**	**As a user, I can access thewebsite any device.**	**High**	**I can access myaccount in any device**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-4	**Technical support**	**USN-18**	**As a user, I can get a customer care support onthe website when am having any queries**	**Medium**	**Helps me to solvequeries**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-4	**Unit Testing**	**USN-19**	**As a user, I can access thewebsite without any interruptions.**	**High**	**I can access the website without any interruptions**	**Chandru, Rishikumar, Suji, Giriharan, Kesavaraman**
Sprint-4	**Integration Testing**	**USN-20**	**As a user, I can access thewebsite without any interruptions.**	**High**	**I can access the website without any interruptions**	**Chandru, Rishikumar, Suji,**

						Giriharan, Kesavaraman
Sprint-4	System Testing	USN-21	As a user, I can access thewebsite without any interruptions.	High	I can access the website without any interruptions	Chandru, Rishikumar, Suji, Giriharan, Kesavaraman
Sprint-4	Correction	USN-22	As a user, I can access thewebsite without any interruptions.	High	I can access the website without any interruptions	Chandru, Rishikumar, Suji, Giriharan, Kesavaraman
Sprint-4	Acceptance Testing	USN-23	As a user, I can access thewebsite without any interruptions.	High	I can access the website without any interruptions	Chandru, Rishikumar, Suji, Giriharan, Kesavaraman
			Submission of Sprint 4			

Planning & Estimation

6.2 Sprint delivery schedule

Since sprints take place over a fixed period of time, it's critical to avoid wasting time during planning and development. And this is precisely where sprint scheduling enters the equation.

In case you're unfamiliar, a sprint schedule is a document that outlines sprint planning from end to end. It's one of the first steps in the agile sprint planning process—and something that requires adequate research, planning, and communication.

Teams often run into trouble when they create more than a few schedules. This can create conflict and derail projects midway through their cycles. To ensure things stay on track, one schedule makes sense.

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**

Velocity:

We have a 6-day sprint duration, and the velocity of the team is 20 (points per sprint). Let'scalculate the team's average velocity (AV) per iteration unit **(story points per day)**

AV = sprint duration / Velocity

AV = 20 / 6

AV = 3.33 SPRINTDELIVERY SCHEDULE

6.3 Reports from JIRA:

Average Age Report.
Created vs Resolved Issues Report.
Pie Chart Report.
Recently Created Issues Report.
Resolution Time Report.
Single Level Group by Report.
Time Since Issues Report.
Time Tracking Report.

7.CODING & SOLUTIONING

Feature 1:

Deployment

```
apiVersion: app/v1
kind: Deployment
metadata:
    name: jobs
spec:
  replicas:1
    selector:
        matchLabels:
        app:flasknode

    template:
     metadata:
        labels:
         app:flasknode
     spec:
       containers:
       - name: flasknode
         images: icr.io/jobtest/jobportalapp
         imagesPullPolicy: Always
         ports:
         -containerPort : 5000
```

Feature 2:

Dockerfile

```
FROM python:3.6
WORKDIR /app
ADD . /app
COPY requirements.txt /app
RUN python3 -m pip install -r requirements.txt
RUN python3 -m pip install ibm_db
EXPOSE 5000
CMD ["python","app.py"]
```

Feature 3:

Flask-Service

```
apiVersion: v1
kind: Service
metadata:
  name: flask-app-service
spec:
  selector:
    app: flask-app
  ports:
  - name: http
    protocol: TCP
    port: 80
    targetPort: 5000
  type: LoadBalancer
```

Feature 4:

Manifest

```
applications:
 - name: Python Flask App IBCMR 2022-10-19
   random-route: true
   memory: 512M
   disk_quoat: 1.5G
```

Feature 5:

Requirement

Flask
ibm_db
sendgrid

TESTING

Test Cases:

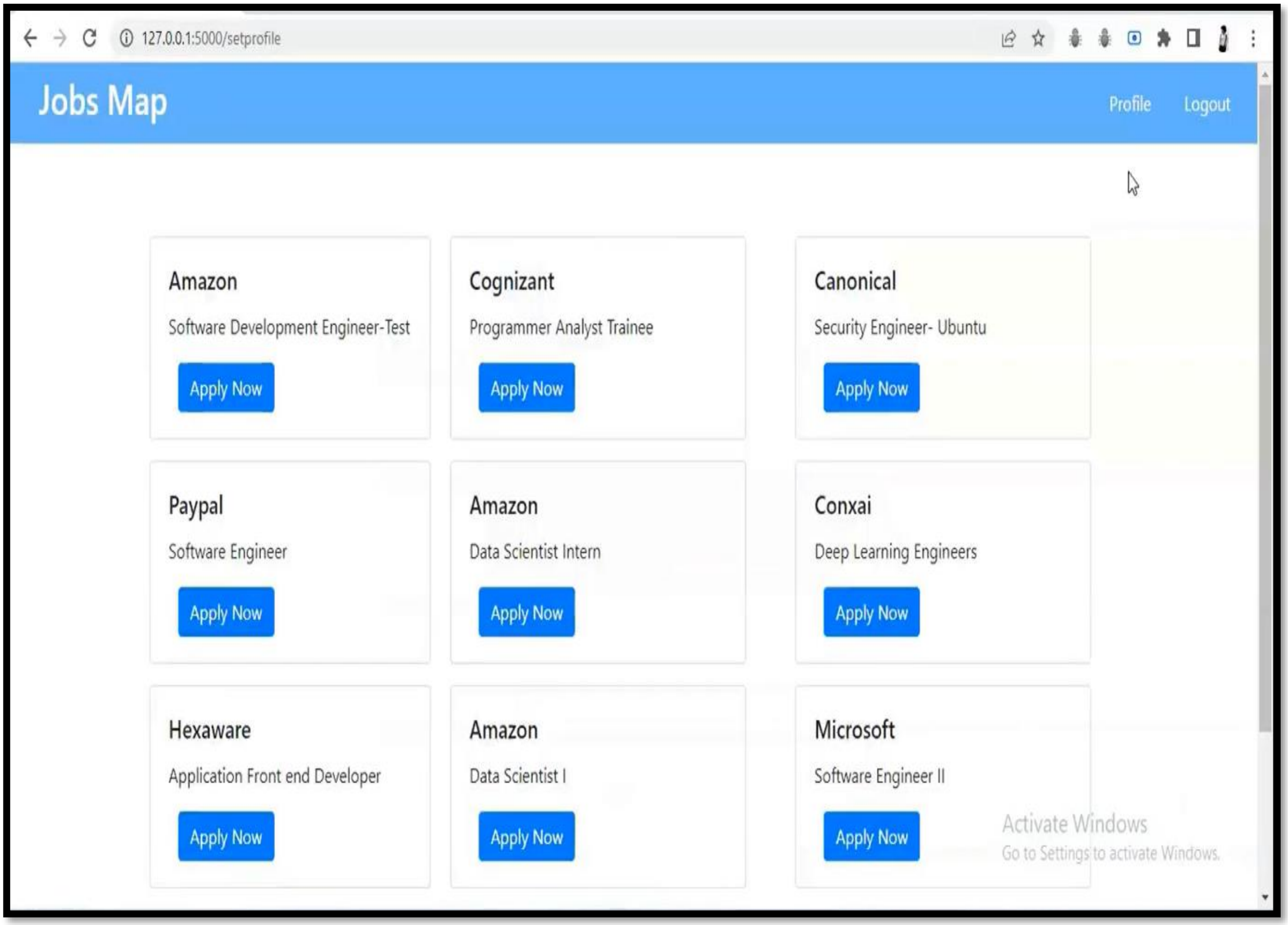
We tested for various validations. Tested all the features with using all the functionalities. Tested the data base storage and retrieval feature too.

Testing was done in phase 1 and phase 2, where issues found in phase1 were fixed and then tested again in phase2.

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.

RESULTS


127.0.0.1:5000/setprofile
Jobs Map
Profile
Logout
Apply Now
Activate Windows
Go to Settings to activate Windows.

Amazon

Software Development Engineer-Test

Cognizant

Programmer Analyst Trainee

Canonical

Security Engineer- Ubuntu

Paypal

Software Engineer

Amazon

Data Scientist Intern

Conxai

Deep Learning Engineers

Hexaware

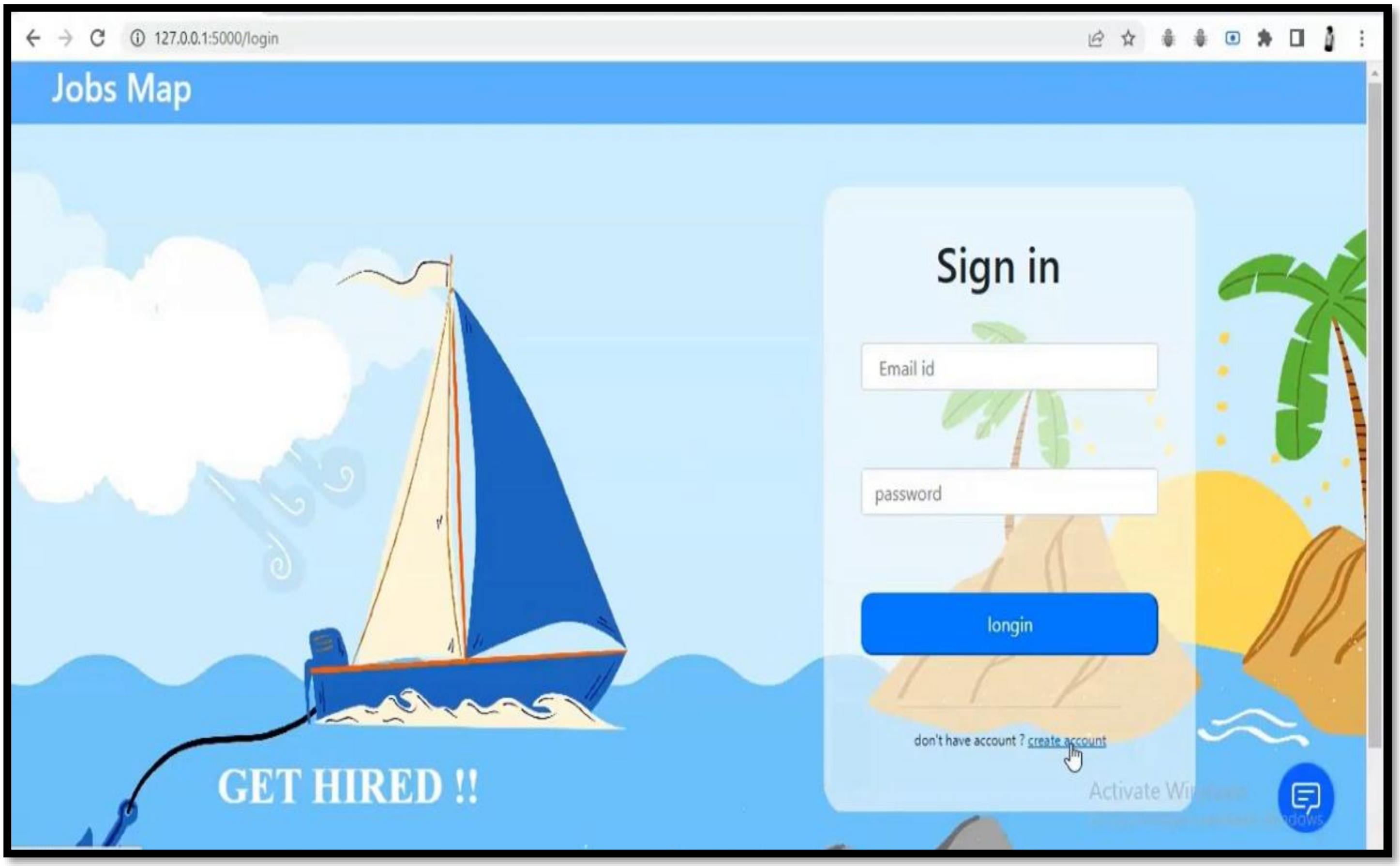
Application Front end Developer

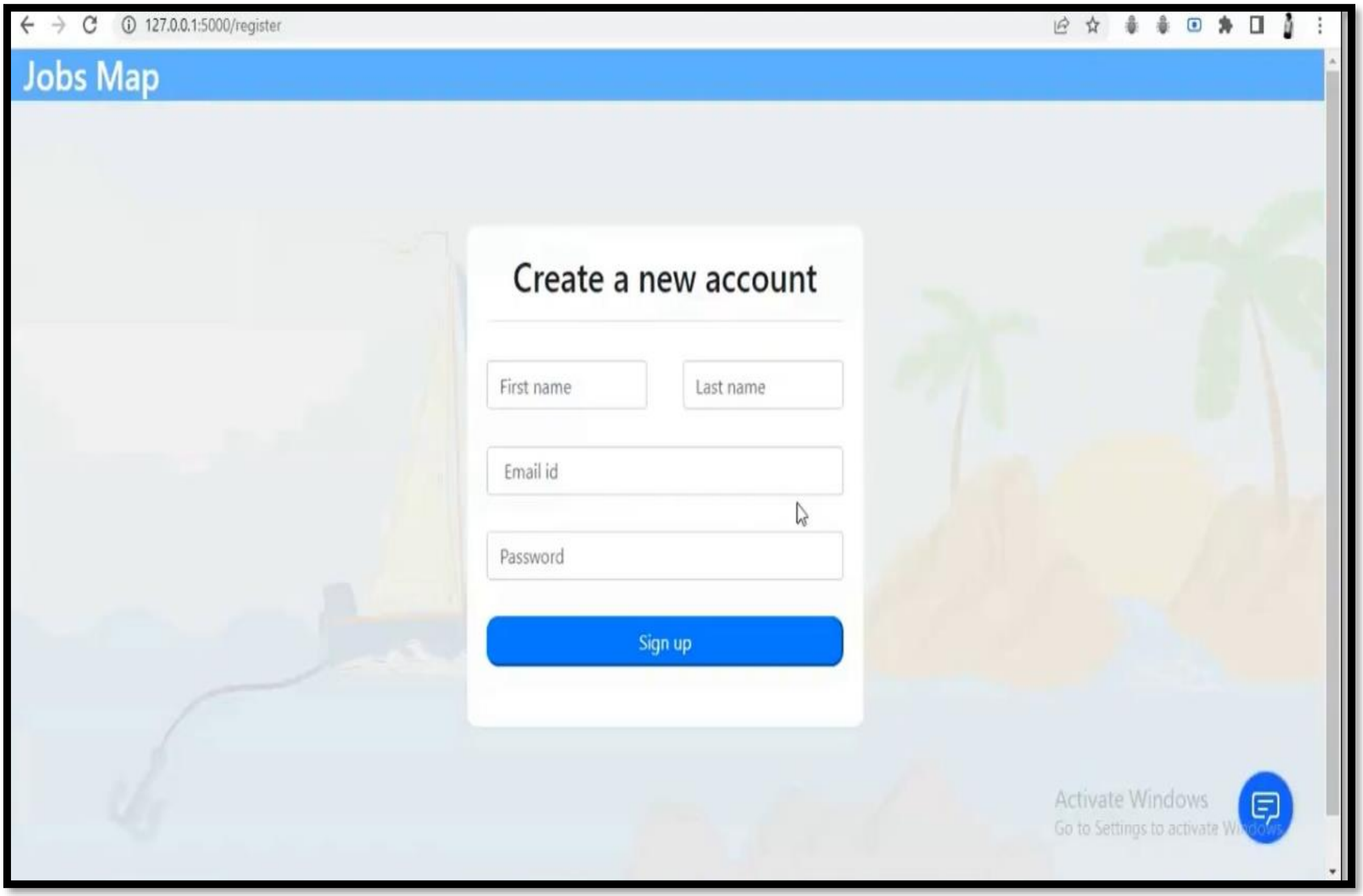
Amazon

Data Scientist I

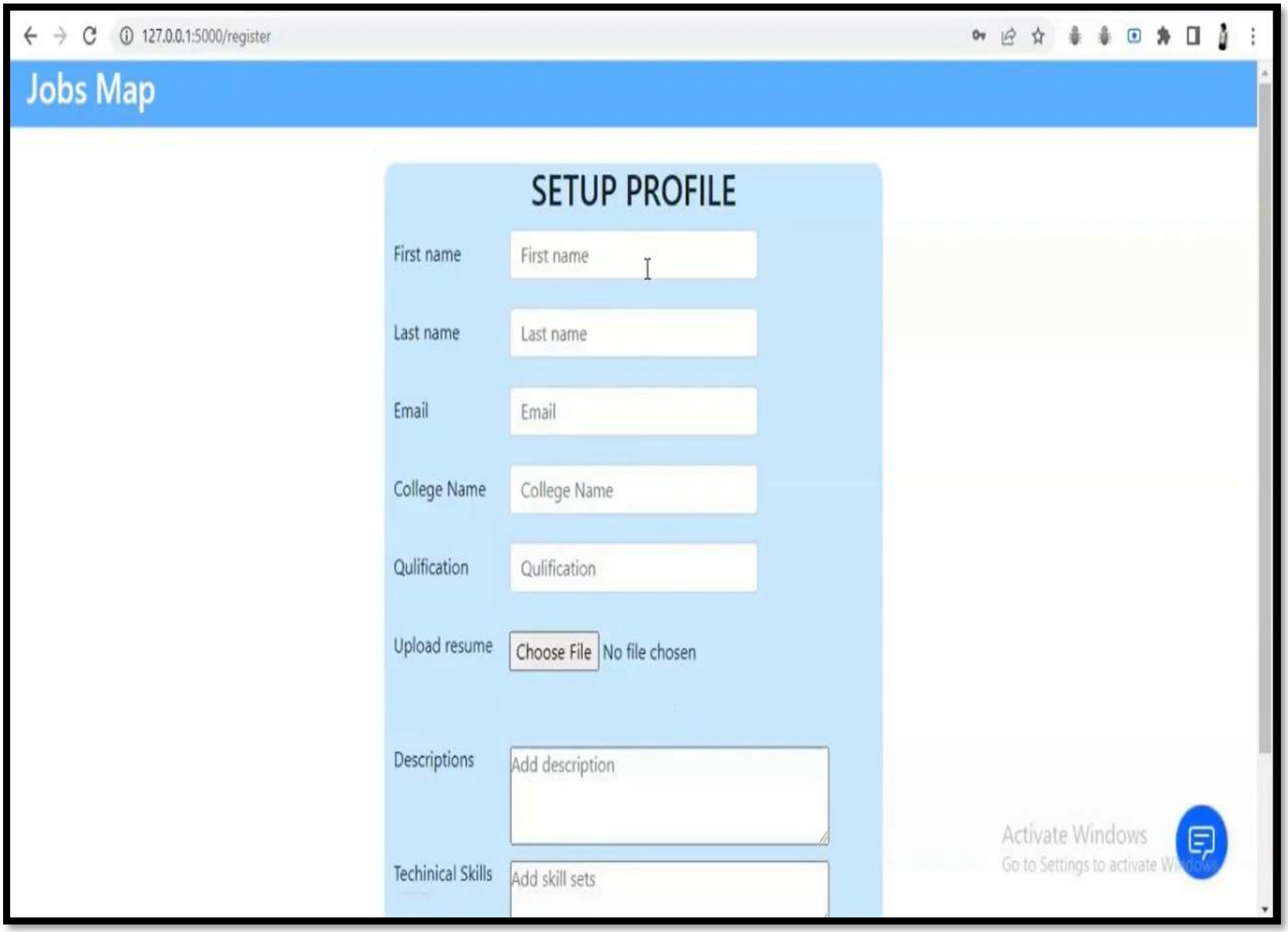
Microsoft

Software Engineer II


127.0.0.1:5000/login
Jobs Map
Sign in
Email id
password
longin
don't have account ? create account
GET HIRED !!
Activate Windows



Create a new account



127.0.0.1:5000/register

Jobs Map

SETUP PROFILE

First name: First name

Last name: Last name

Email: Email

College Name: College Name

Qulification: Qulification

Upload resume: Choose File No file chosen

Descriptions: Add description

Techinical Skills: Add skill sets

ADVANTAGES AND DISAVANTAGES

ADVANTAGE :

a. It helps candidates to search the job which perfectly suites them and makethem aware of all the job openings.
b. It help recruiters of the company to choose the right candidates for theirorganizations with appropriate skills.
c. Since it is cloud application, it does require any installation of softwareand is portable.

DISADVANTAGE:

d. It is costly.
e. Uninterrupted internet connection is required for smooth functioning ofapplication

CONCLUSION

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

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 techniquesto recommend data in a efficient way.

APPENDIX

SOURCE CODE

Login Page:

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>My Page Title</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/style.css') }}">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">


  </head>
  <body >
    <div class="tit">
      <h2>Jobs Map</h2>
      </div>
<form class="login-form" action="/dashbord" method="get">


  <div class="body-img">

  <div class="">
  <div class="heading">
      <P class="title-txt"><span class=".search"></span> <br><span class="for"></span> <br>
<span class="job"> </span>  </P>
  </div>
  <div class="botton-heading">
      <P class="bottom-txt">GET HIRED  !!</P>
  </div>
    <div class="login-box">
      <div class="title-box">
        <h1>Sign in</h1>
      </div>
      <div class="mail-box">
        <input class="txt-box form-control" type="email" name="username" placeholder=" Email id"
title="Email or user name">
      </div>
      <div class="password">
        <input class="txt-box form-control" type="password" name="password"
placeholder="password" title="password">
      </div>
      <div class="btu-login">
        <button class="log-btn btn-primary"type="login" name="login"title="login">longin</button>

      </div>
      <hr class="line">
      <div class="create-acc">
         <p>don't have account ? <a href="register" title="create new account">create
account</a><span>{{ msg }}</span>

         </p>
      </div>
    </div>
  </div>
  </div>




</form>
<script>
window.watsonAssistantChatOptions = {
integrationID: "1e37e3f8-b996-4518-935d-5d45ea53a543", // The ID of this integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "6a00d3de-c85d-45d1-ac93-3662219a5b2b", // The ID of your service instance.
onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script>
```

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

Set Profile

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>SET ACCOUNT</title>
      <link rel="stylesheet" 
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N" 
crossorigin="anonymous">


          <link rel="stylesheet" href="{{ url_for('static', filename='css/setaccount.css') }}">
<link rel="icon" type="image/x-icon" href="images/favicon.ico">
  </head>
  <body>
    <div class="tit">
<h2 style="color:white;margin-left:-87%"><a href="/login" style="text-
decoration:none;color:white">Jobs Map</a></h2>
    </div>
    <div class="msg">{{ msg }}</div>
<form class="setAccount" action="/setprofile" method="post" enctype="multipart/form-data">
  <div class="profile">
    <h2>SETUP PROFILE</h2>
    <div class="name">

      <div class="firstname">
        <label>First name</label>
        <input class="fname txt-box form-control" type="text" name="firstname" placeholder="First 
name" title="First name">
      </div>

      <div class="lastname">
          <label>Last name</label>
        <input class="lname txt-box form-control" type="text" name="lastname" placeholder="Last 
name" title="First name">
      </div>
    </div>
    <div class="email">
        <label>Email</label>
      <input class="email-box txt-box form-control" type="email" name="email" title="Email"
placeholder="Email">
    </div>
    <div class="college">
  <label>College Name</label>
  <input class="clgname txt-box form-control" type="text" name="college" placeholder="College
Name">
    </div>
    <div class="education">
        <label>Qulification</label>
        <input class="Qulification  txt-box form-control"type="text" name="qulification"
placeholder="Qulification">



</div>

    <div class="resume">
      <label>Upload resume </label>

        <input class="resume-Upload"type="file" name="resume"  title="Upload resume">


    </div>
    <div class="desciption">
      <label>Descriptions</label>
      <textarea class="decription-box" name="decription" rows="3" cols="60" placeholder="Add
description"></textarea>

    </div>
    <div class="desciption">
      <label>Techinical Skills</label>
      <textarea class="decription-box" name="skills" rows="2" cols="60" placeholder="Add skill
sets"></textarea>
    </div>

    <div class="intrest">
      <label>Filed of intrest</label>
      <select class="interest-opt" name="interest">
        <option value="Select">select</option>
      <option value="web development">web developmen</option>
      <option value="Cloud computing">Cloud computing</option>
      <option value="Data sciencet">Data science</option>
      <option value="Data analytics">Data analytics</option>
      <option value="Blockchin">Blockchin</option>
     </select>

    </div>



    <div class="next">
     <button class ="next-btn  btn-primary"type="submit" name="button">Finish  </button>
    </div>




  </div>

</form>
<script>
window.watsonAssistantChatOptions = {
integrationID: "1e37e3f8-b996-4518-935d-5d45ea53a543", // The ID of this integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "6a00d3de-c85d-45d1-ac93-3662219a5b2b", // The ID of your service instance.
onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script>

  </body>
</html>
```

Register Page

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

```
  <head>
    <meta charset="utf-8">
    <title>My Page Title</title>
    <link rel="icon" type="image/x-icon" href="images/favicon.ico">
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css" integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">


      <link rel="stylesheet" href="{{ url_for('static', filename='css/register.css') }}">
  </head>
  <body>
    <div class="tit">

<h2 style="color:white;margin-left:-87%"><a href="#" style="text-
decoration:none;color:white">Jobs Map</a></h2>
    </div>
    <form class="" action="/register" method="post">

      <div class="body-img">

            <div class="register-box">
              <div class="re-Title">
                <h2>Create a new account</h2>
                <hr>
              </div>
              <div class="lastFirst-name">
                <div class="first-name">
                  <input class="txt-box form-control" type="text" name="firstname" placeholder="First
name" title="First name">
                </div>
                <div class="last-name">
                  <input class="txt-box form-control" type="text" name="lastname" placeholder="Last
name" title="Last name">
                </div>
              </div>

              <div class="email">

                <input class="txt-box form-control" type="email" name="email" placeholder=" Email id"
title="Email ">
              </div>
              <div class="password">
                  <input class="txt-box form-control" type="password" name="password"
placeholder="Password" title="password">
              </div>


              <div class="signup">
            <button class="signup-btn btn-primary"type="submit" name="login"title="login">Sign
up</button>
              </div>
            </div>
      </div>


    </form>
<div class="msg">{{ msg }}</div>
<script>
window.watsonAssistantChatOptions = {
integrationID: "1e37e3f8-b996-4518-935d-5d45ea53a543", // The ID of this integration.
region: "au-syd", // The region your integration is hosted in.
serviceInstanceID: "6a00d3de-c85d-45d1-ac93-3662219a5b2b", // The ID of your service instance.
onLoad: function(instance) { instance.render(); }
};
setTimeout(function(){
const t=document.createElement('script');
t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" +
(window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
document.head.appendChild(t);
});
</script>
  </body>
</html>
```

Display Page

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="{{ url_for('static', filename='css/display.css') }}">
<link rel="icon" type="image/x-icon" href="images/favicon.ico">
    <title></title>
  </head>
  <body>
    <div class="tit">
      <div class="heading">
        <h2><a href="#">Jobs Map</a></h2>
      </div>

      <div class="row ">
        <a  class="btn "  href="/dashbord">FIND JOBS </a>
        <a  class="btn " href="/logout">LOGOUT</a>
      </div>
    </div>
    <div class="details-box">
      <h2><u>Your details</u></h2>

      <table class="tab-box"><br><br><br><br>
        <tr>
          <td>firstname:</td>
          <td>Prasath</td>
        </tr>
        <tr>
          <td>lastname:</td>
          <td>JS</td>
        </tr>
        <tr>
          <td>Email ID:</td>
          <td>rishikumar@gmail.com</td>
        </tr>
        <tr>
          <td>College:</td>
          <td>ABC college</td>
        </tr>
        <tr>
```

```
     <td>Qualification:</td>
     <td>BE-CSE</td>
    </tr>
    <tr>
     <td>Skills:</td>
     <td>JAVA,C++</td>
    </tr>
    <tr>
     <td>Intrest:</td>
     <td>Web Development</td>
    </tr>
  </div>
  <script>
 window.watsonAssistantChatOptions = {
  integrationID: "1e37e3f8-b996-4518-935d-5d45ea53a543", // The ID of this integration.
  region: "au-syd", // The region your integration is hosted in.
  serviceInstanceID: "6a00d3de-c85d-45d1-ac93-3662219a5b2b", // The ID of your service instance.
  onLoad: function(instance) { instance.render(); }
 };
 setTimeout(function(){
  const t=document.createElement('script');
  t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
  document.head.appendChild(t);
 });
</script>
 </body>
</html>
```