PNT2022TMID28997

IBM – NALAIYA THIRAN PROJECT

SKILL JOB RECOMMENDER APPLICATION

ABSTRACT

In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personalized job suggestions. 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 contributions 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 recommendation based on professional skills of job seekers; and iii) carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. We thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue.

There has been a sudden boom in the technical industry and an increase in the number of good startups. 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.

TABLE OF CONTENT

1. INTRODUCTION

1.1 PROJECT OVERVIEW

There has been a sudden boom in the technical industry and an increase in the number of good startups. 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 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

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.

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.

2.2 REFERENCES:

1. 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
2. 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.
3. 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.
4. 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.
5. 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.
6. 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.
7. T Mikolov et al. “Efficient estimation of word representations in vector space”. In: arXiv preprint arXiv:1301.3781 (2013).
8. 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

"Can an efficient recommender system be modelled 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 recruitment s 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 & PROPOSED SOLUTION

In this project you will be working on two modules:

1. Admin and
2. User

ADMIN:

The role of the admin is to check out the database about the stock and have a track of all the things that the users are purchasing.

USER:

The user will login into the website and go through the products available on the website. Instead of navigating to several screens the user can directly talk to Chatbot. Get the recommendations based on information provided by the user.

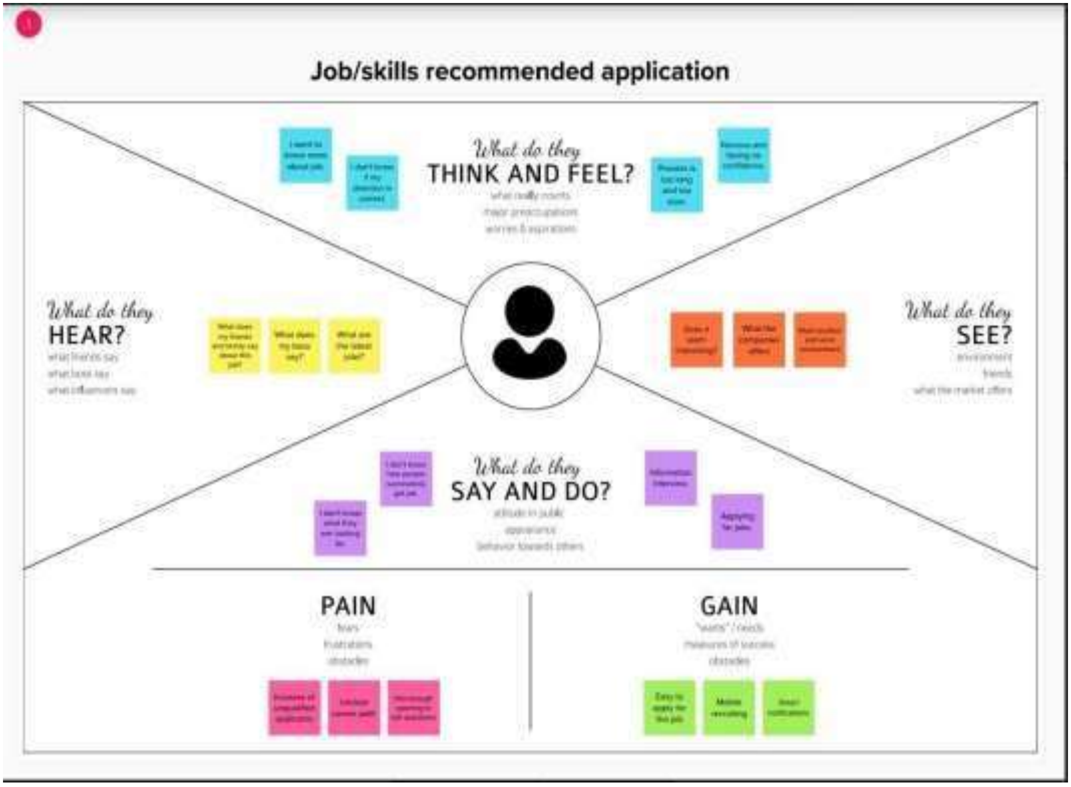
FEATURES OF CHATBOT:

- Using chatbot we can manage user's choices and orders.
- The chatbot can give recommendations to the users based on their interests.
- It can promote the best deals and offers on that day.
- It will store the customer's details and orders in the database.
- The chatbot will send a notification to customers if the order is confirmed.
- Chatbots can also help in collecting customer feedback.

3.1 EMPATHY MAP CANVAS:

Empathy Map Canvas: An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

example:



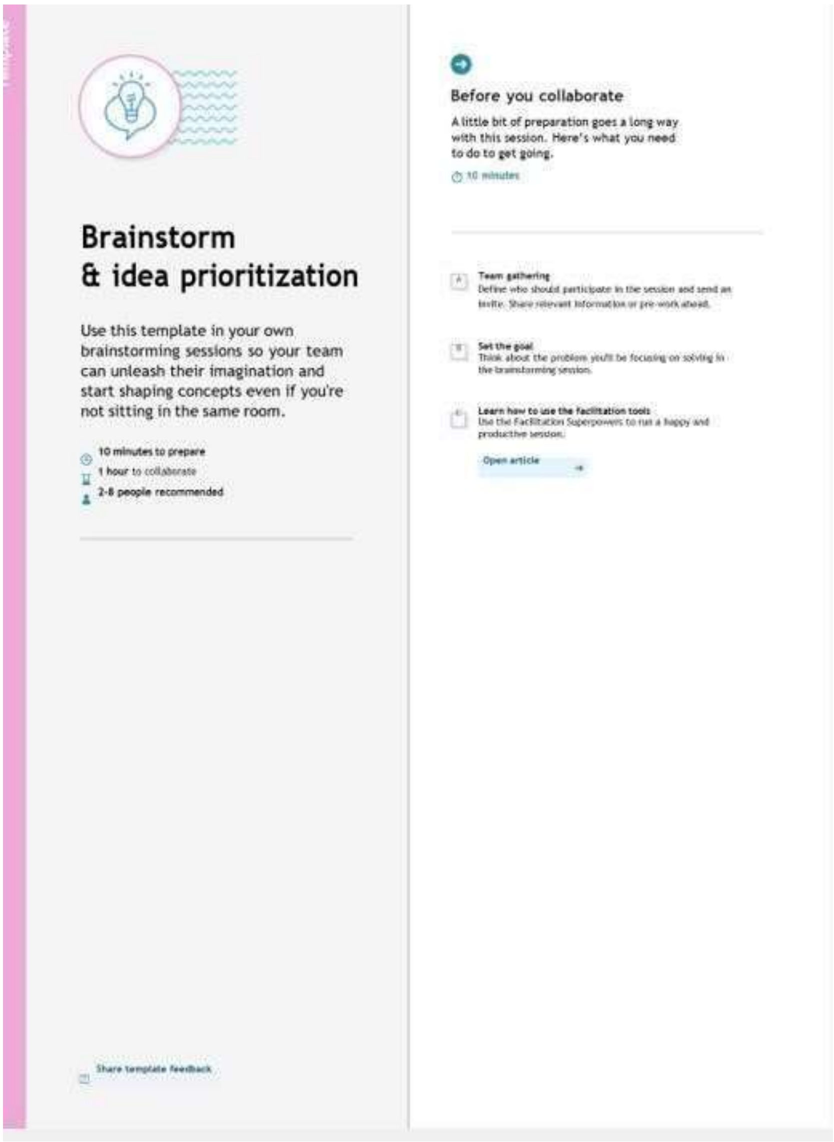
3.2 IDEATION & BRAINSTROMING:

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 number 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

Share template feedback

Before you collaborate

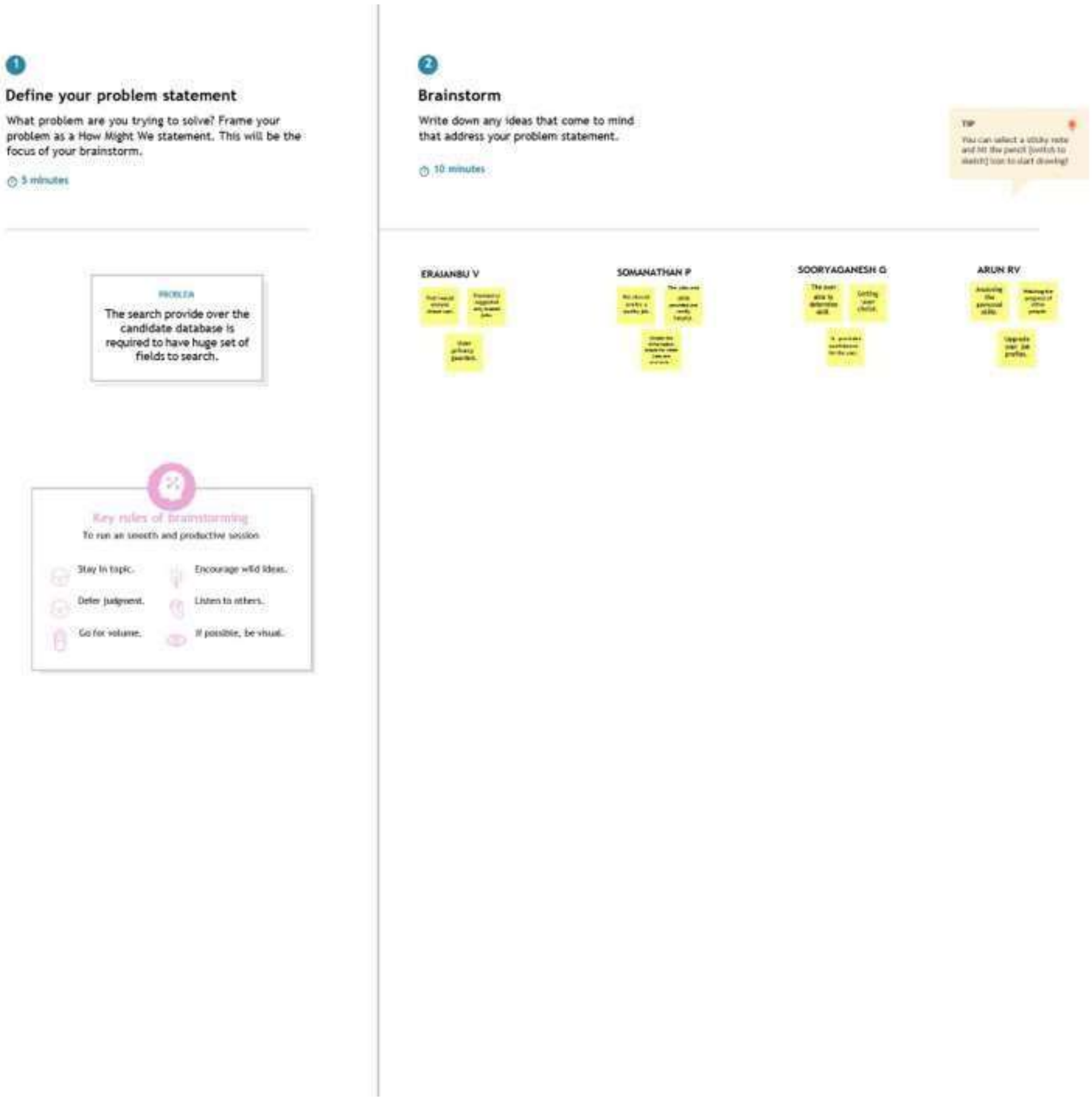
A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

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

The search provide over the candidate database is required to have huge set of fields to search.

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.

Encourage wild ideas.

Defer judgment.

Listen to others.

Go for volume.

If possible, be visual.

2

Brainstorm

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!

ERAIANBU V

[illegible]

[illegible]

User privacy guarded.

SOMANATHAN P

[illegible]

[illegible]

[illegible]

SOORYAGANESH G

The user able to determine skill.

Getting user choice.

It provides confidence for the user.

ARUN RV

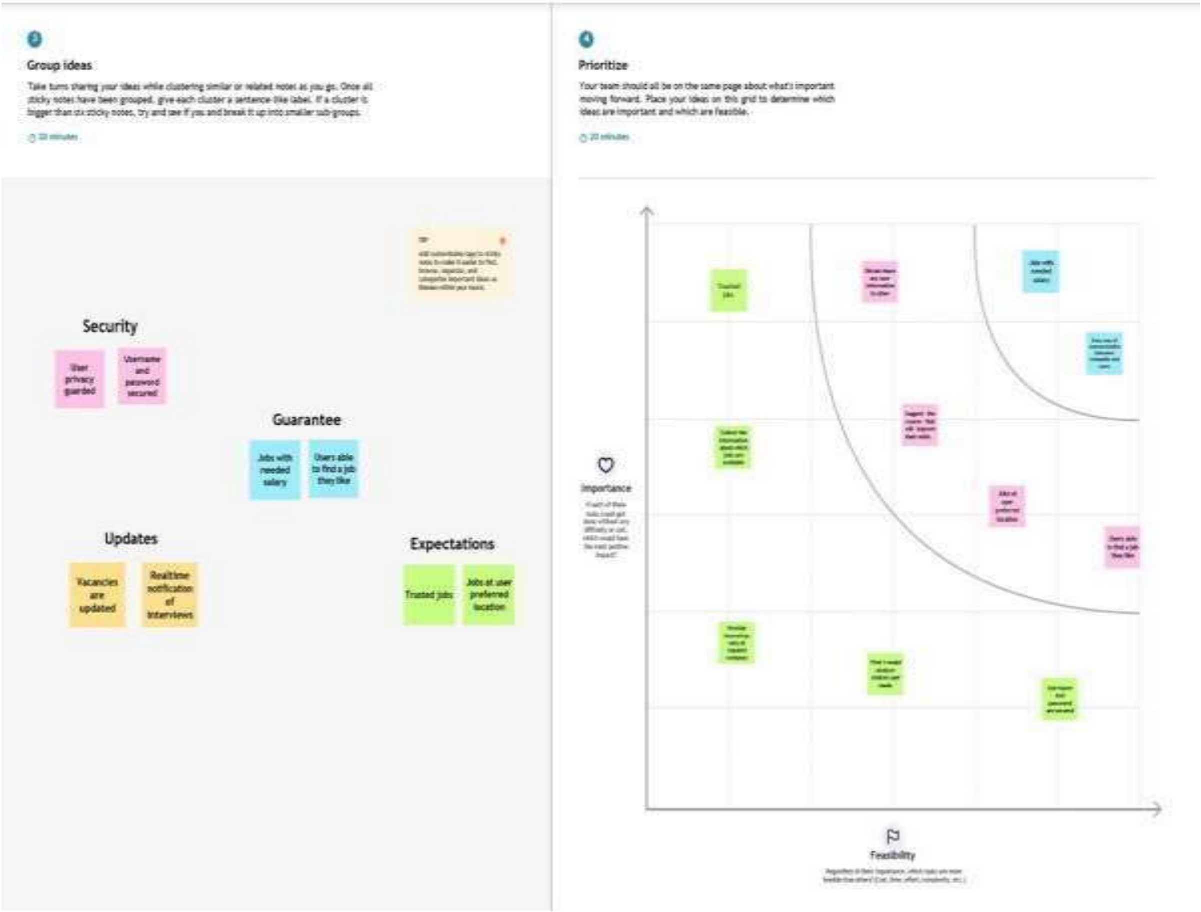
Analysing the personal skills.

[illegible]

Upgrade user job profiles.

STEP 2:

Brainstorm, Idea Listing and Grouping



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

Security
User privacy guarded
Username and password secured
Guarantee
Jobs with needed salary
Users able to find a job they like
Updates
Vacancies are updated
Realtime notification of interviews
Expectations
Trusted jobs
Jobs at user preferred location

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

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.

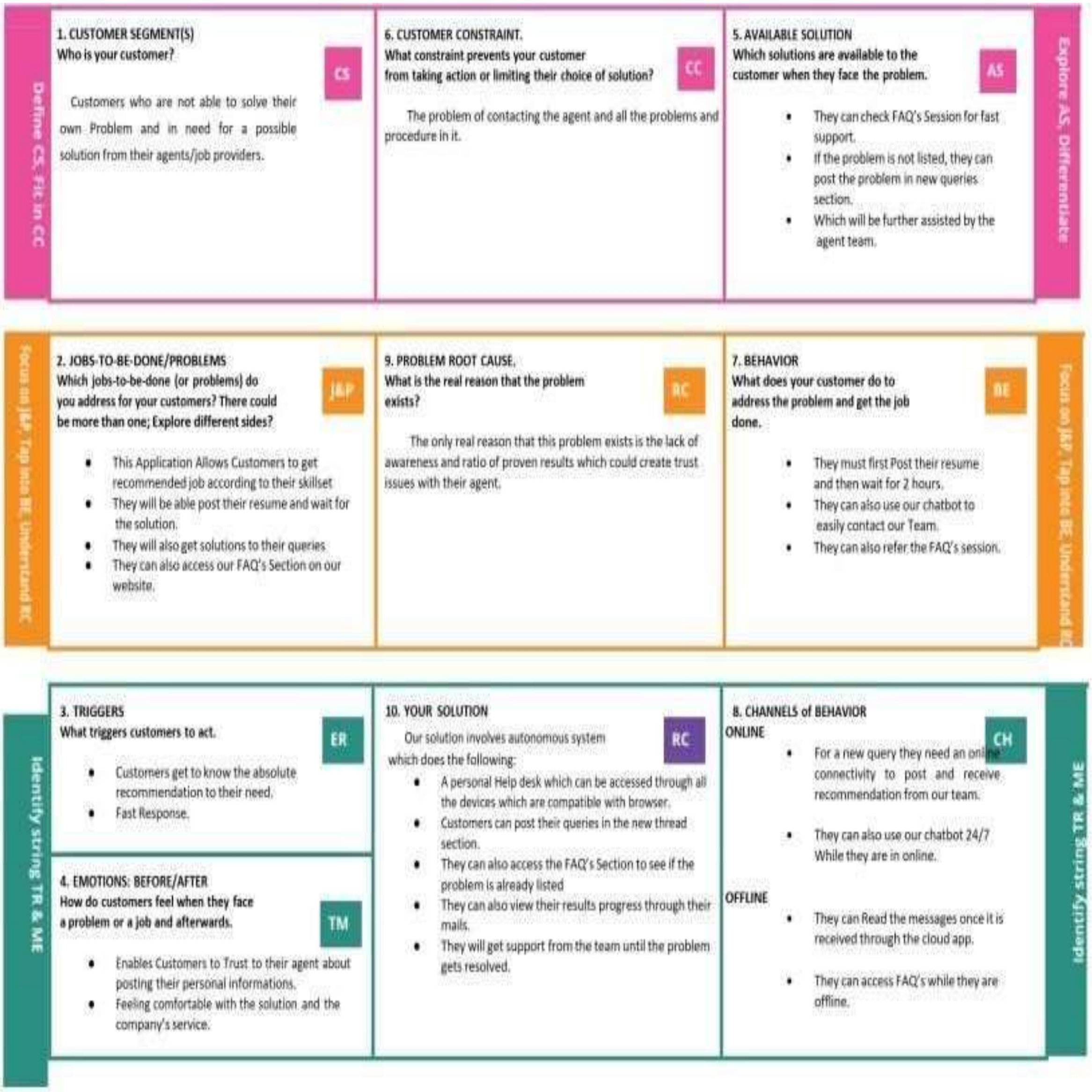
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities and threats (SWOT) to develop a plan.

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 in CC
Explore AS, Differentiate
Focus on J&P, Tap into BE, Understand RC
Focus on J&P, Tap into BE, Understand RC
Identify string TR & ME
Identify string TR & ME

1. CUSTOMER SEGMENT(S)
Who is your customer?

CS

Customers who are not able to solve their own Problem and in need for a possible solution from their agents/job providers.

6. CUSTOMER CONSTRAINT.
What constraint prevents your customer from taking action or limiting their choice of solution?

CC

The problem of contacting the agent and all the problems and procedure in it.

5. AVAILABLE SOLUTION
Which solutions are available to the customer when they face the problem.

AS

- They can check FAQ's Session for fast support.
- If the problem is not listed, they can post the problem in new queries section.
- Which will be further assisted by the agent team.

2. JOBS-TO-BE-DONE/PROBLEMS
Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; Explore different sides?

J&P

- This Application Allows Customers to get recommended job according to their skillset
- They will be able post their resume and wait for the solution.
- They will also get solutions to their queries
- They can also access our FAQ's Section on our website.

9. PROBLEM ROOT CAUSE.
What is the real reason that the problem exists?

RC

The only real reason that this problem exists is the lack of awareness and ratio of proven results which could create trust issues with their agent.

7. BEHAVIOR
What does your customer do to address the problem and get the job done.

BE

- They must first Post their resume and then wait for 2 hours.
- They can also use our chatbot to easily contact our Team.
- They can also refer the FAQ's session.

3. TRIGGERS
What triggers customers to act.

ER

- Customers get to know the absolute recommendation to their need.
- Fast Response.

4. EMOTIONS: BEFORE/AFTER
How do customers feel when they face a problem or a job and afterwards.

TM

- Enables Customers to Trust to their agent about posting their personal informations.
- Feeling comfortable with the solution and the company's service.

10. YOUR SOLUTION

RC

Our solution involves autonomous system which does the following:

- A personal Help desk which can be accessed through all the devices which are compatible with browser.
- Customers can post their queries in the new thread section.
- They can also access the FAQ's Section to see if the problem is already listed
- They can also view their results progress through their mails.
- They will get support from the team until the problem gets resolved.

8. CHANNELS of BEHAVIOR

CH

ONLINE

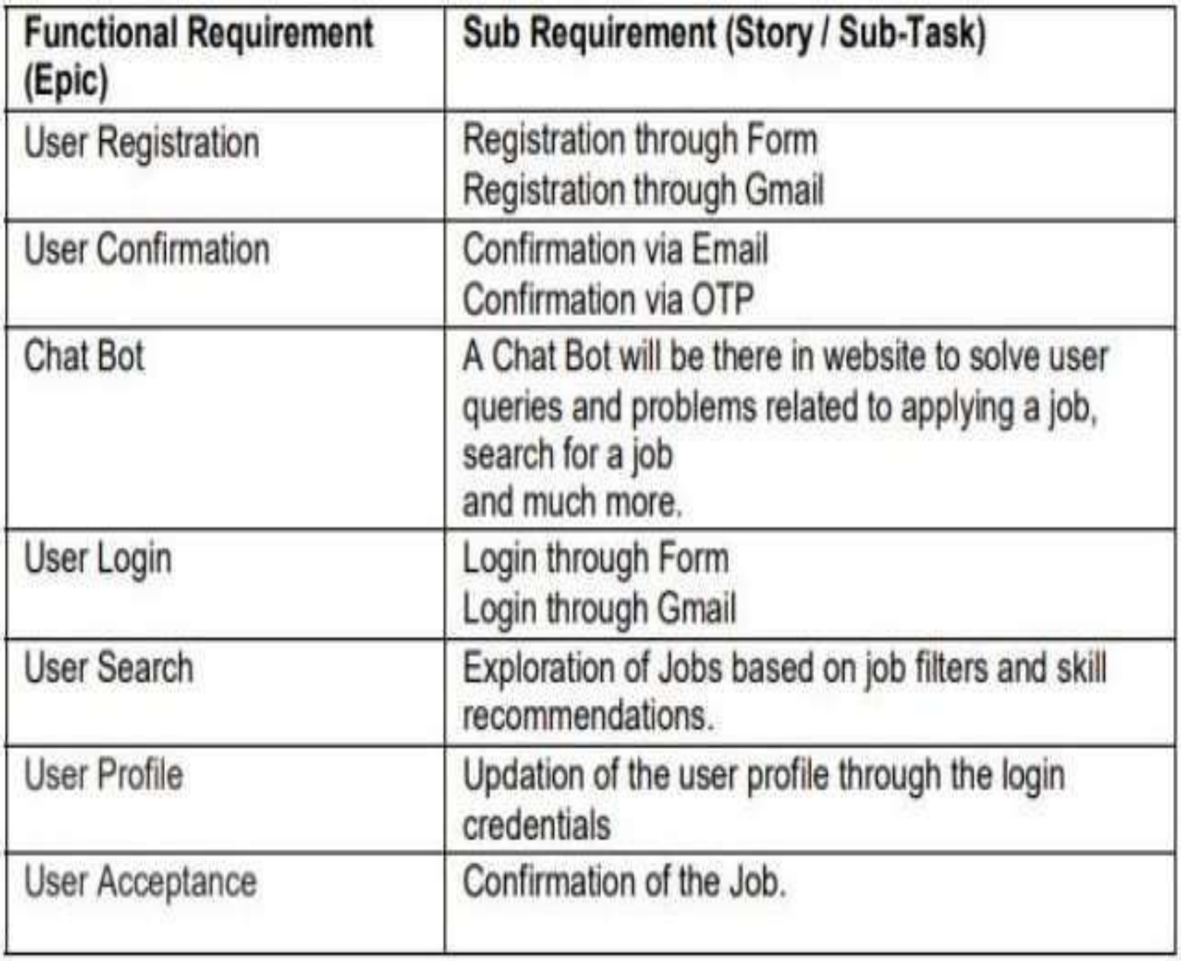
- For a new query they need an online connectivity to post and receive recommendation from our team.
- They can also use our chatbot 24/7 While they are in online.

OFFLINE

- They can Read the messages once it is received through the cloud app.
- They can access FAQ's while they are offline.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:



Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
User Registration	Registration through Form Registration through Gmail
User Confirmation	Confirmation via Email Confirmation via OTP
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.
User Login	Login through Form Login through Gmail
User Search	Exploration of Jobs based on job filters and skill recommendations.
User Profile	Updation of the user profile through the login credentials
User Acceptance	Confirmation of the Job.

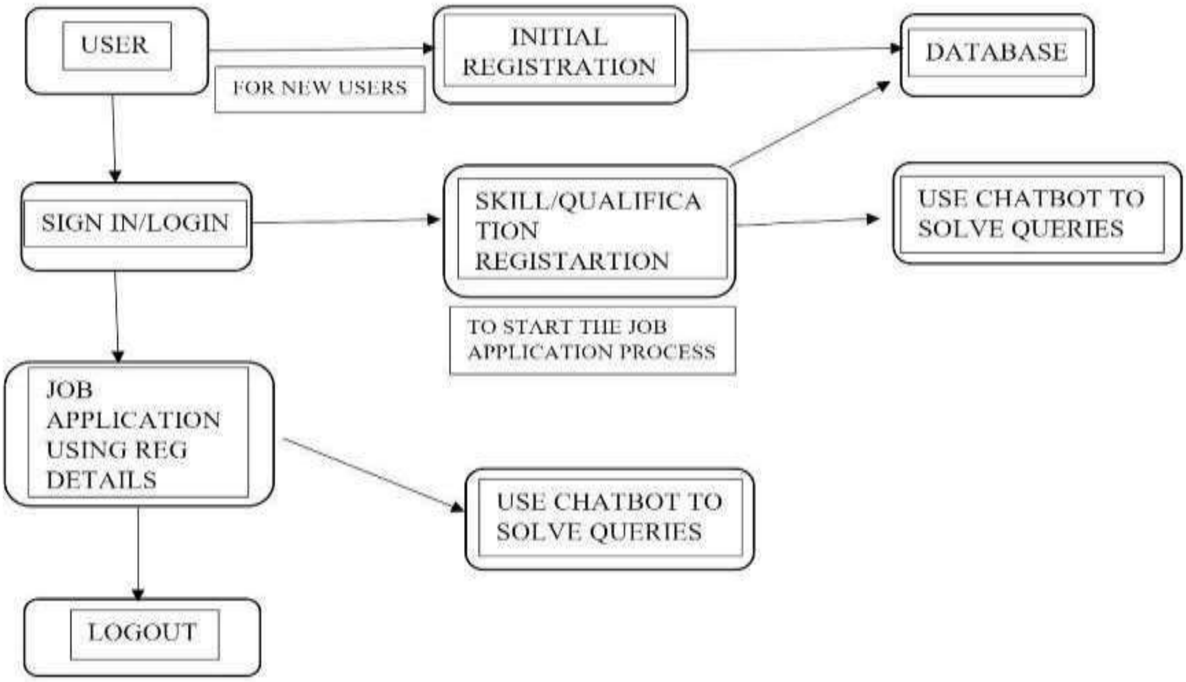
4.2 NON-FUNCTIONAL REQUIREMENTS:

1. Usability
2. Security
3. Reliability
4. Performance
5. Availability
6. Scalability

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

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


USER
FOR NEW USERS
INITIAL REGISTRATION
DATABASE
SIGN IN/LOGIN
SKILL/QUALIFICA TION REGISTARTION
USE CHATBOT TO SOLVE QUERIES
TO START THE JOB APPLICATION PROCESS
JOB APPLICATION USING REG DETAILS
USE CHATBOT TO SOLVE QUERIES
LOGOUT

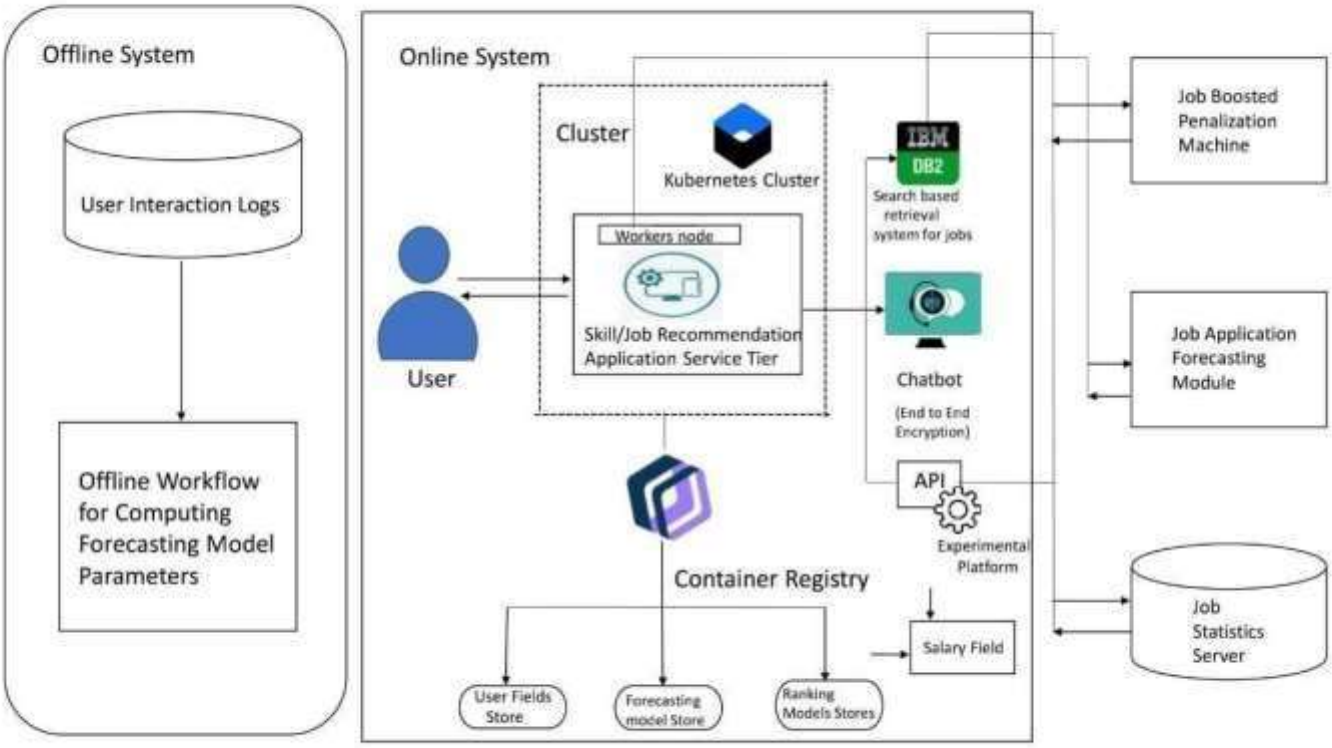
5.2 SOLUTION & TECHNICAL ARCHITECTURE:

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

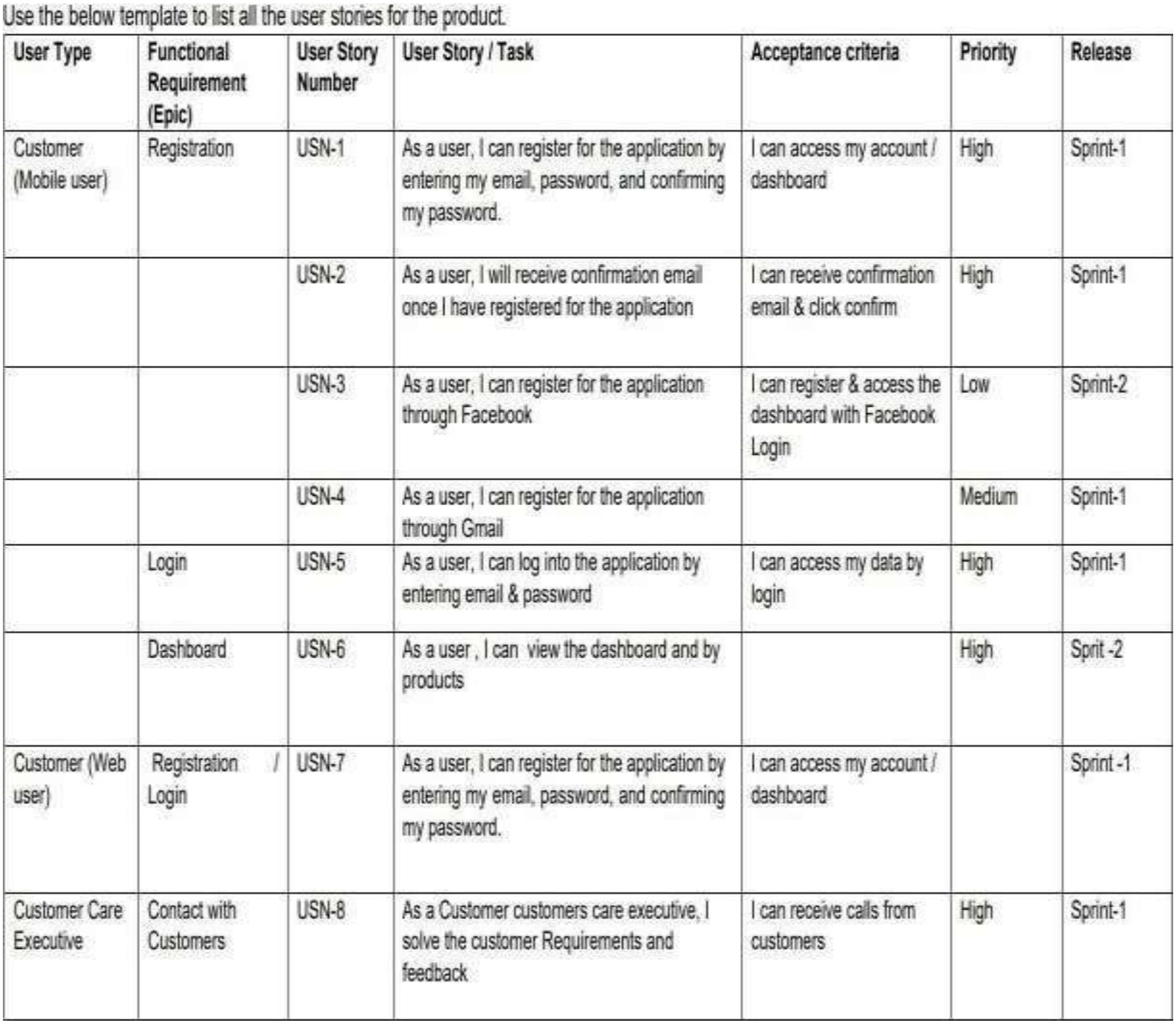
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution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

- Provide the best business require recommend by using the optimised and efficient algorithm

- Differentiate the fake job recommend by fake sites and be aware from the Scammers

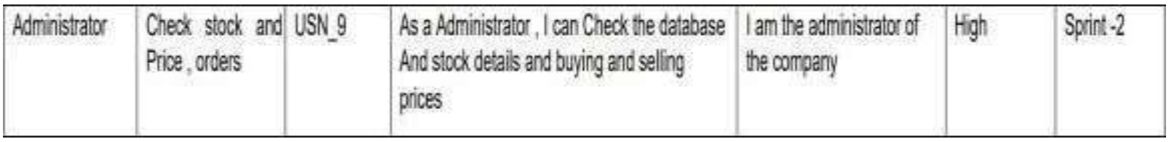


5.3 USER STORIES:



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

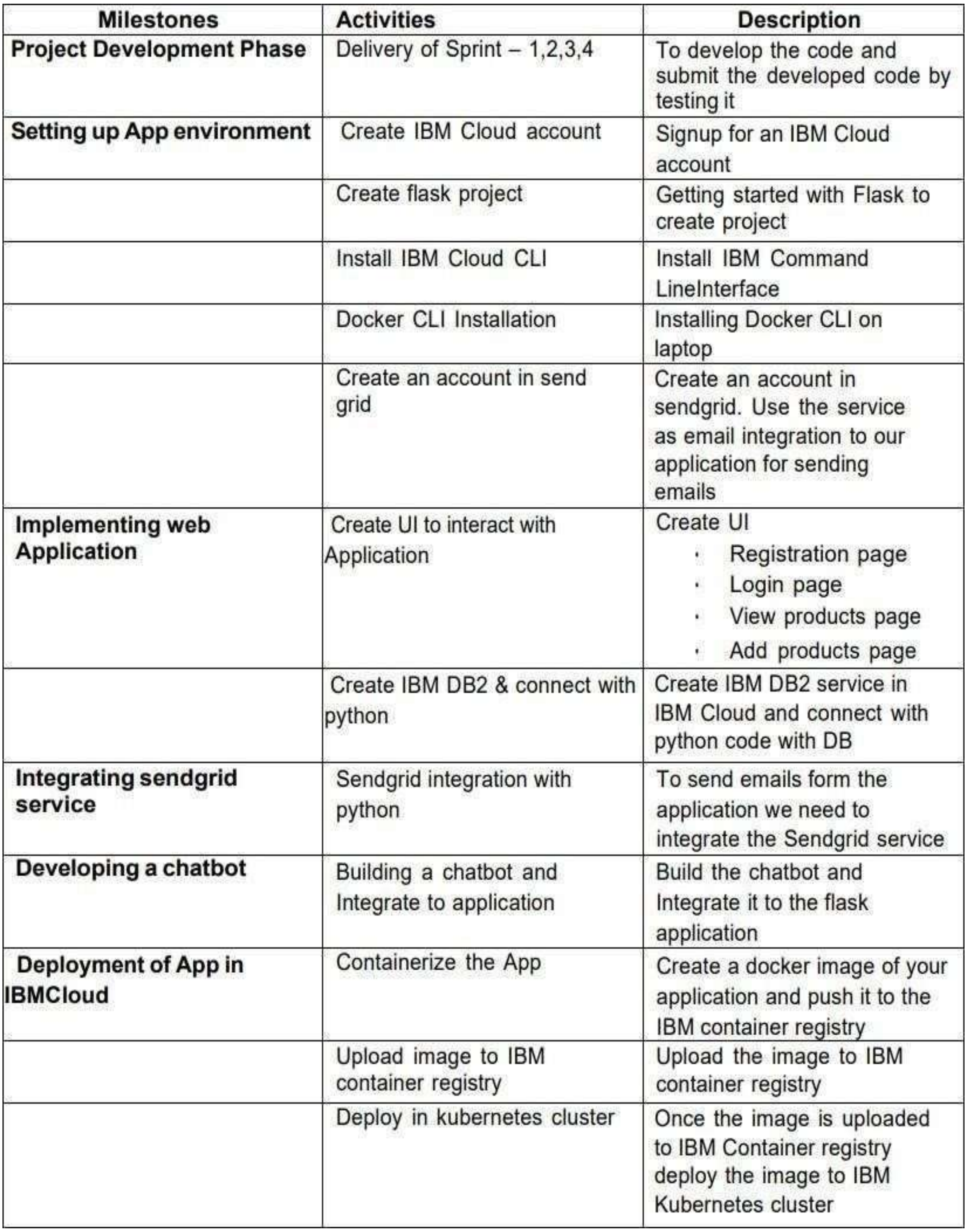
User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	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	I can access my data by login	High	Sprint-1
	Dashboard	USN-6	As a user , I can view the dashboard and by products		High	Sprit -2
Customer (Web user)	Registration / Login	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard		Sprint -1
Customer Care Executive	Contact with Customers	USN-8	As a Customer customers care executive, I solve the customer Requirements and feedback	I can receive calls from customers	High	Sprint-1



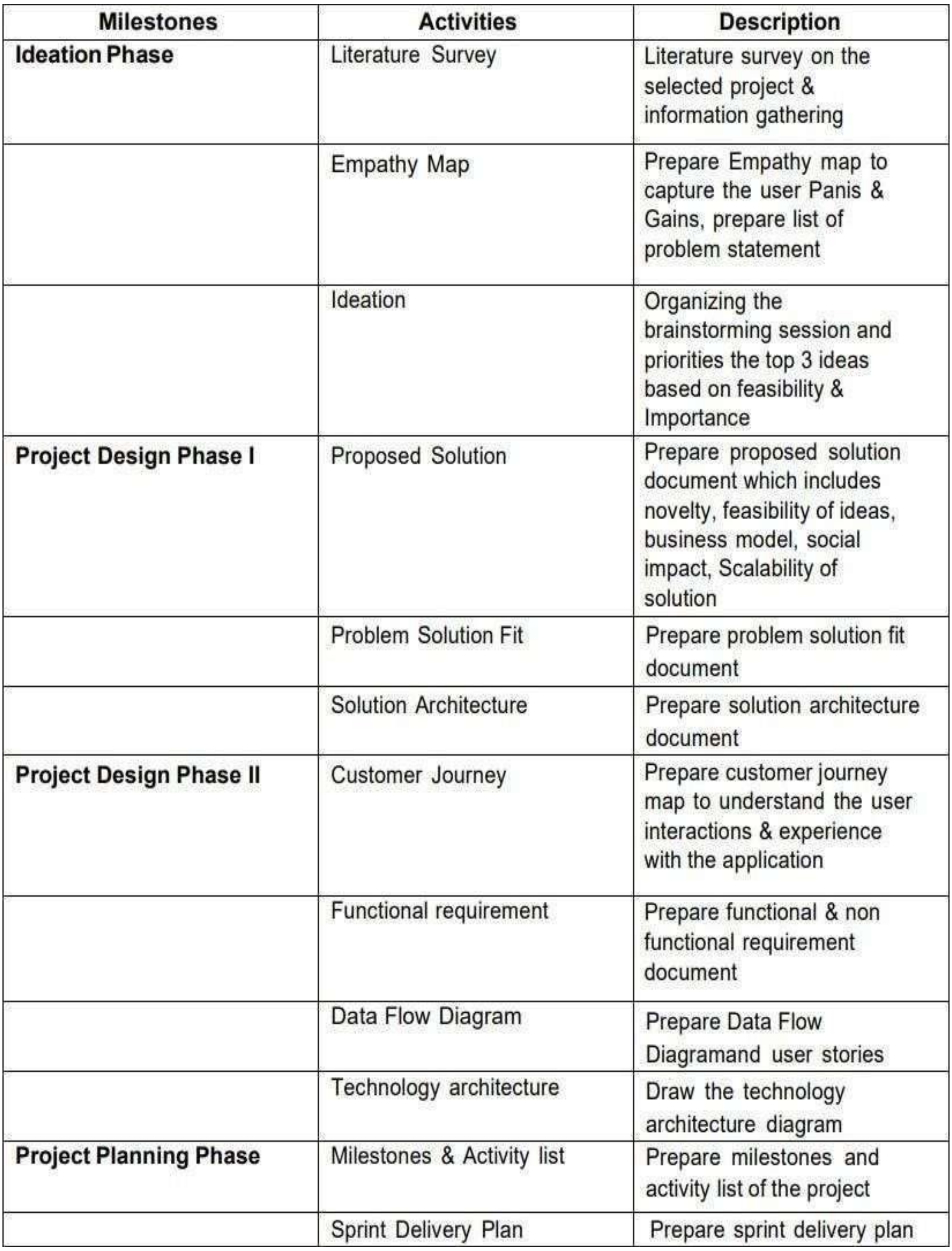
Administrator	Check stock and Price , orders	USN_9	As a Administrator , I can Check the database And stock details and buying and selling prices	I am the administrator of the company	High	Sprint -2

6. PROJECT PLANNING & SCHEDULE

6.1 SPRINT PLANNING & ESTIMATION:

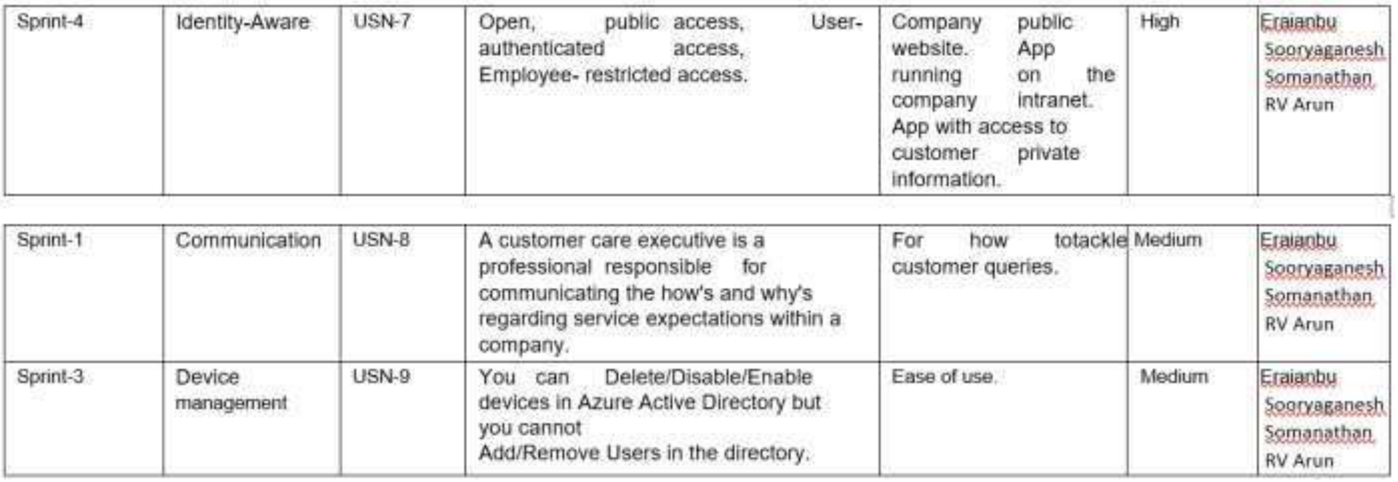


Milestones	Activities	Description
Project Development Phase	Delivery of Sprint – 1,2,3,4	To develop the code and submit the developed code by testing it
Setting up App environment	Create IBM Cloud account	Signup for an IBM Cloud account
	Create flask project	Getting started with Flask to create project
	Install IBM Cloud CLI	Install IBM Command LineInterface
	Docker CLI Installation	Installing Docker CLI on laptop
	Create an account in send grid	Create an account in sendgrid. Use the service as email integration to our application for sending emails
Implementing web Application	Create UI to interact with Application	Create UI · Registration page · Login page · View products page · Add products page
	Create IBM DB2 & connect with python	Create IBM DB2 service in IBM Cloud and connect with python code with DB
Integrating sendgrid service	Sendgrid integration with python	To send emails form the application we need to integrate the Sendgrid service
Developing a chatbot	Building a chatbot and Integrate to application	Build the chatbot and Integrate it to the flask application
Deployment of App in IBMCloud	Containerize the App	Create a docker image of your application and push it to the IBM container registry
	Upload image to IBM container registry	Upload the image to IBM container registry
	Deploy in kubernetes cluster	Once the image is uploaded to IBM Container registry deploy the image to IBM Kubernetes cluster

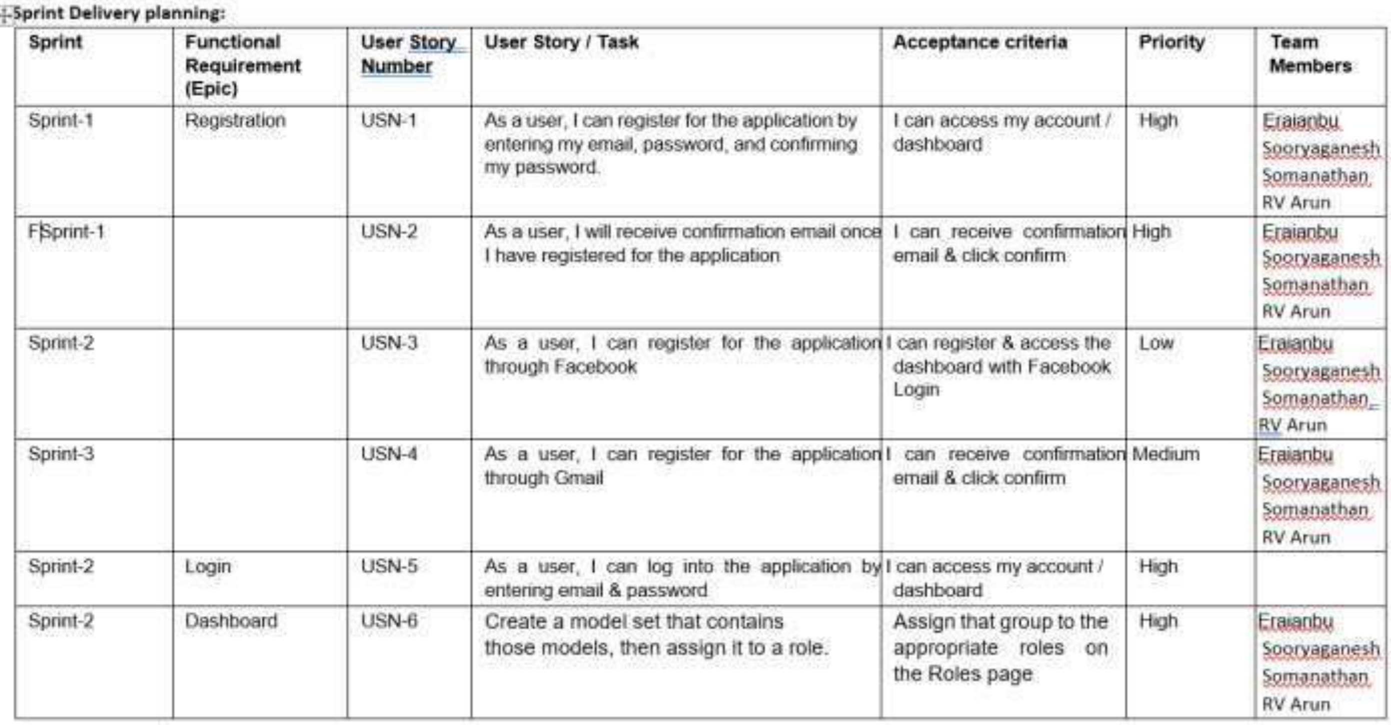


Milestones	Activities	Description
Ideation Phase	Literature Survey	Literature survey on the selected project & information gathering
	Empathy Map	Prepare Empathy map to capture the user Panis & Gains, prepare list of problem statement
	Ideation	Organizing the brainstorming session and priorities the top 3 ideas based on feasibility & Importance
Project Design Phase I	Proposed Solution	Prepare proposed solution document which includes novelty, feasibility of ideas, business model, social impact, Scalability of solution
	Problem Solution Fit	Prepare problem solution fit document
	Solution Architecture	Prepare solution architecture document
Project Design Phase II	Customer Journey	Prepare customer journey map to understand the user interactions & experience with the application
	Functional requirement	Prepare functional & non functional requirement document
	Data Flow Diagram	Prepare Data Flow Diagramand user stories
	Technology architecture	Draw the technology architecture diagram
Project Planning Phase	Milestones & Activity list	Prepare milestones and activity list of the project
	Sprint Delivery Plan	Prepare sprint delivery plan

6.2 SPRINT DELIVERY SCHEDULE:

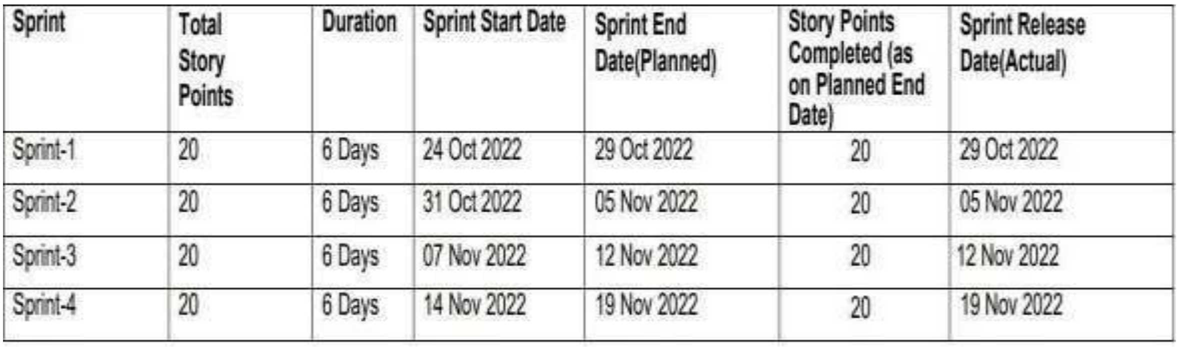


Sprint-4	Identity-Aware	USN-7	Open, public access, User-authenticated access, Employee- restricted access.	Company public website. App running on the company intranet. App with access to customer private information.	High	Eraianbu Sooryaganesh Somanathan RV Arun
Sprint-1	Communication	USN-8	A customer care executive is a professional responsible for communicating the how's and why's regarding service expectations within a company.	For how totackle customer queries.	Medium	Eraianbu Sooryaganesh Somanathan RV Arun
Sprint-3	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the directory.	Ease of use.	Medium	Eraianbu Sooryaganesh Somanathan RV Arun

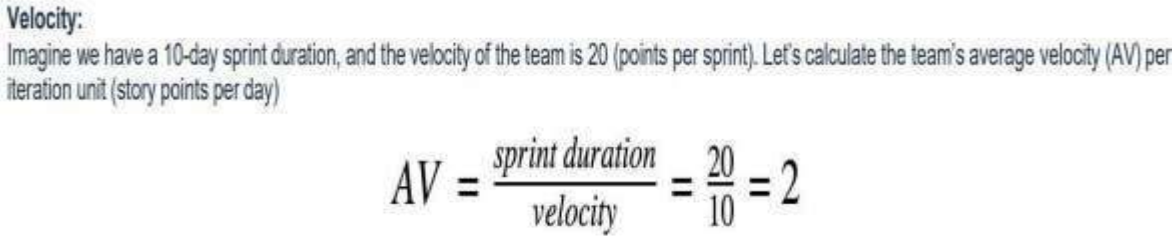


Sprint Delivery planning:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Members
Sprint-1	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	Eraianbu Sooryaganesh Somanathan RV Arun
FSprint-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	Eraianbu Sooryaganesh Somanathan RV Arun
Sprint-2		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Eraianbu Sooryaganesh Somanathan_ RV Arun
Sprint-3		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Eraianbu Sooryaganesh Somanathan RV Arun
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	
Sprint-2	Dashboard	USN-6	Create a model set that contains those models, then assign it to a role.	Assign that group to the appropriate roles on the Roles page	High	Eraianbu Sooryaganesh Somanathan RV Arun



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

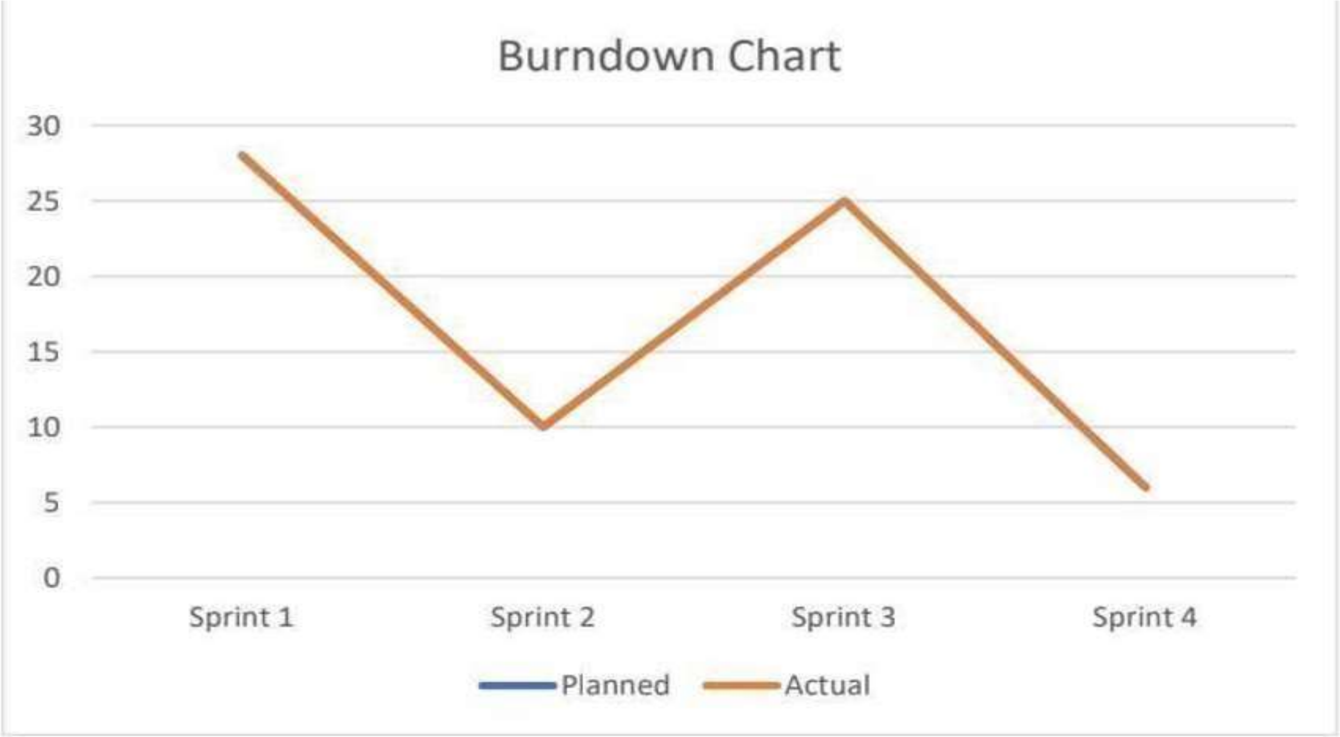


Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 REPORTS FROM JIRA:



7. CODING & SOLUTIONING

7.1 FEATURE-1:

INDEX.HTML:

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

```
import bcrypt

import ibm_db

from sendmail import *

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


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




app = Flask(__name__)

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


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

def home():

    if 'email' not in session:

      return redirect(url_for('index'))

    return render_template('index.html',name='Home')

@app.route("/index")

def index():

  return render_template('index.html')

@app.route("/index1")

def index1():

  return render_template('index1.html')


@app.route("/job_details")

def  job_details():

  return render_template('job_details.html')
```

```
@app.route("/job_details1")
def  job_details1():
  return render_template('job_details1.html')

@app.route("/job_details2")
def  job_details2():
   return render_template('job_details2.html')

@app.route("/job_details3")
def  job_details3():


  return render_template('job_details3.html')

@app.route("/job_details4")
def  job_details4():
  return render_template('job_details4.html')

@app.route("/job_details5")
def job_details5():
  return render_template('job_details5.html')

@app.route("/job_details6")
def job_details6():
  return render_template('job_details6.html')

@app.route("/job_listing")
def job_listing():
  return render_template('job_listing.html')

@app.route("/about")
def about():
  return render_template('about.html')
@app.route("/registeration",methods=['GET','POST'])
def register():
  if request.method == 'POST':
    name = request.form['name']
    phn = request.form['phn']
    email = request.form['email']
    psw = request.form['psw']


    if not name or not email or not phn or not psw:
      return render_template('registeration.html',error='Please fill all fields')
    hash=bcrypt.hashpw(psw.encode('utf-8'),bcrypt.gensalt())
    query = "SELECT * FROM user_detail WHERE email=? OR phn=?"
    stmt = ibm_db.prepare(conn, query)
    ibm_db.bind_param(stmt,1,email)
    ibm_db.bind_param(stmt,2,phn)
    ibm_db.execute(stmt)
    print(stmt)
    isUser = ibm_db.fetch_assoc(stmt)
    if not isUser:
      insert_sql = "INSERT INTO user_detail(name, email, phn, psw) VALUES (?,?,?,?)"
      prep_stmt = ibm_db.prepare(conn, insert_sql)
      ibm_db.bind_param(prep_stmt, 1, name)
      ibm_db.bind_param(prep_stmt, 2, email)
      ibm_db.bind_param(prep_stmt, 3, phn)
      ibm_db.bind_param(prep_stmt, 4, hash)
      ibm_db.execute(prep_stmt)


      sendMailUsingSendGrid(API,from_email,to_emails,subject,html_content)
      return render_template('registeration.html',success="You can login")
    else:
      return render_template('registeration.html',error='Invalid Credentials')

  return render_template('registeration.html',name='Home')

@app.route("/login",methods=['GET','POST'])
def login():
    if request.method == 'POST':
      email = request.form['email']
      psw = request.form['psw']

      if not email or not psw:
        return render_template('login.html',error='Please fill all fields')
      query = "SELECT * FROM user_detail WHERE email=?"
      stmt = ibm_db.prepare(conn, query)
      ibm_db.bind_param(stmt,1,email)
      ibm_db.execute(stmt)
      isUser = ibm_db.fetch_assoc(stmt)
      print(isUser,psw)

      if not isUser:
        return render_template('login.html',error='Invalid Credentials')
      isPasswordMatch = bcrypt.checkpw(psw.encode('utf-8'),isUser['PSW'].encode('utf-8'))

      if not isPasswordMatch:
        return render_template('login.html',error='Invalid Credentials')

      session['email'] = isUser['EMAIL']
      return redirect(url_for('home'))

    return render_template('login.html',name='Home')
```

```
@app.route("/apply",methods=['GET','POST'])
def apply():
 if request.method == 'POST':
  name = request.form['name']
  email = request.form['email']
  psw = request.form['password']
  age = request.form['age']
  job = request.form['job']
  interest = request.form['interest']

  if not name or not email or not psw:
   return render_template('apply.html',error='Please fill all fields')
  hash=bcrypt.hashpw(psw.encode('utf-8'),bcrypt.gensalt())
  query = "SELECT * FROM applyform WHERE email=? OR psw=?"
  stmt = ibm_db.prepare(conn, query)
  ibm_db.bind_param(stmt,1,email)
  ibm_db.bind_param(stmt,2,psw)
  ibm_db.execute(stmt)
  isUser = ibm_db.fetch_assoc(stmt)
  if not isUser:
   insert_sql = "INSERT INTO admin_detail(name, email, psw,age,job,interest) VALUES (?,?,?,?,?,?)"

   prep_stmt = ibm_db.prepare(conn, insert_sql)
   ibm_db.bind_param(prep_stmt, 1, name)
   ibm_db.bind_param(prep_stmt, 2, email)
   ibm_db.bind_param(prep_stmt, 3, psw)
   ibm_db.bind_param(prep_stmt, 4, age)
   ibm_db.bind_param(prep_stmt, 5, job)
   ibm_db.bind_param(prep_stmt, 6, interest)
   ibm_db.execute(prep_stmt)
```

```
      return render_template('apply.html',success="You can login")
    else:
      return render_template('apply.html',error='Invalid Credentials')


  return render_template('apply.html',name='Home')



if __name__ == "__main__":
    app.run(debug=True)
```

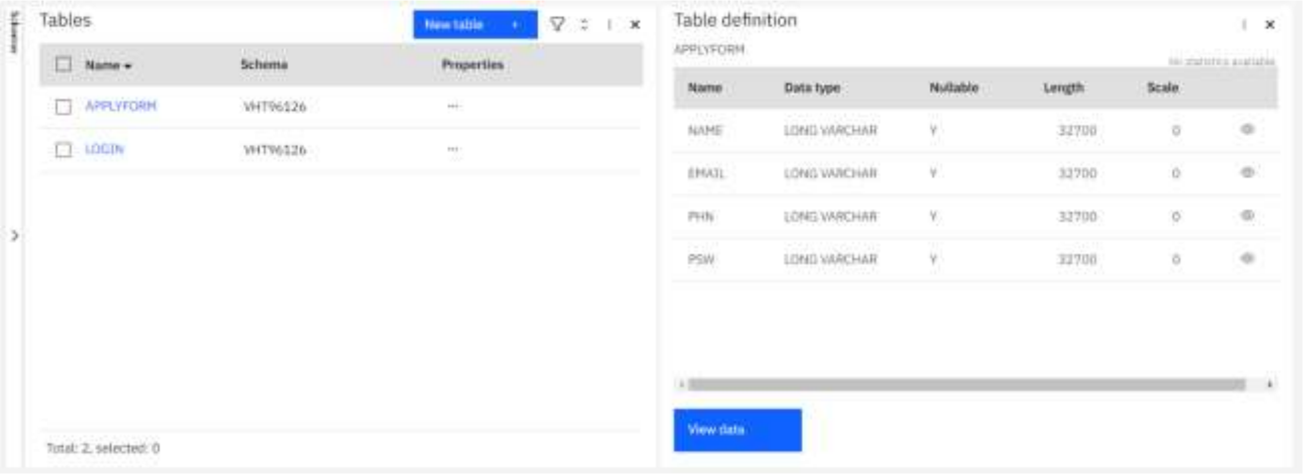
7.2 Feature 2

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

```
<script>
            window.watsonAssistantChatOptions = {
              integrationID: "65c01ed6-9fc1-4883-979a-3676279ebe44", // The ID of this integration.
              region: "us-south", // The region your integration is hosted in.
              serviceInstanceID: "8fcd017f-a192-420a-aafc-18cb0330efca", // The ID of your service instance.
              onLoad: function(instance) { instance.render(); }
            };
            setTimeout(function(){
              const t=document.createElement('script');
              t.src="https://web-chat.global.assistant.watson.appdomain.cloud/versions/" + (window.watsonAssistantChatOptions.clientVersion || 'latest') + "/WatsonAssistantChatEntry.js";
              document.head.appendChild(t);
            });
          </script>
```

7.3 Database Schema (if Applicable)

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



Tables

Name	Schema	Properties
APPLYFORM	VHT96126	...
LOGIN	VHT96126	...

Total: 2, selected: 0

Table definition

APPLYFORM

No statistics available

Name	Data type	Nullable	Length	Scale
NAME	LONG VARCHAR	Y	32700	0
EMAIL	LONG VARCHAR	Y	32700	0
PHN	LONG VARCHAR	Y	32700	0
PSW	LONG VARCHAR	Y	32700	0

8. TESTING

8.1 Test Cases

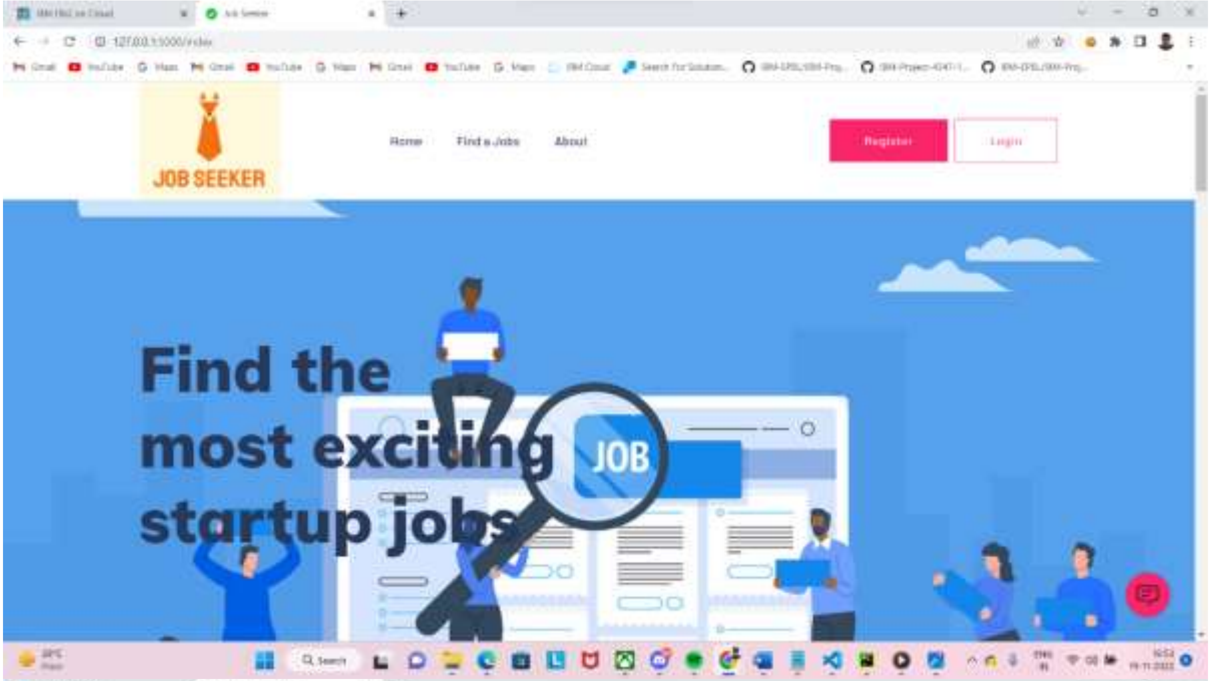
We tested for various validations. Tested all the features with using all thefunctionalities. Tested the data base storage and retrieval feature too. Testing was done in phase 1 and phase 2, where issues found in phase1 werefixed and then tested again in phase2.

8.2 User Acceptance Testing

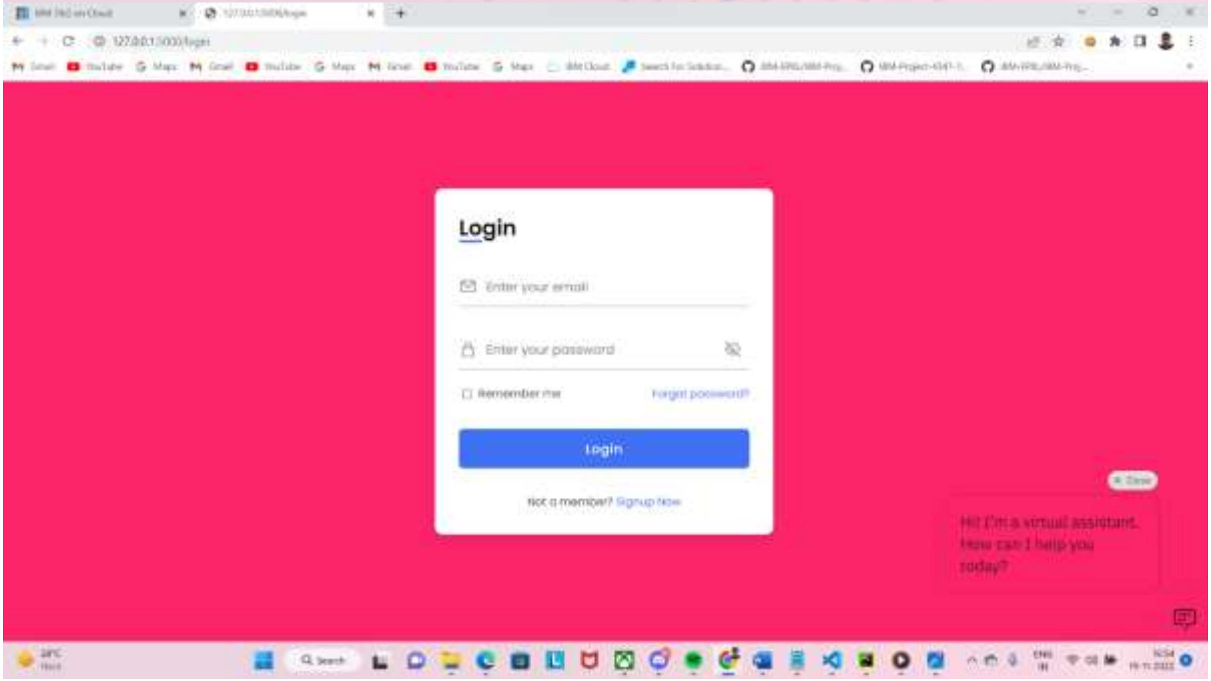
Real world testing was also done, by giving to remote users and asking them touse the application. Their difficulties were fixed and tested again until all the issues were fixed.

9. RESULTS

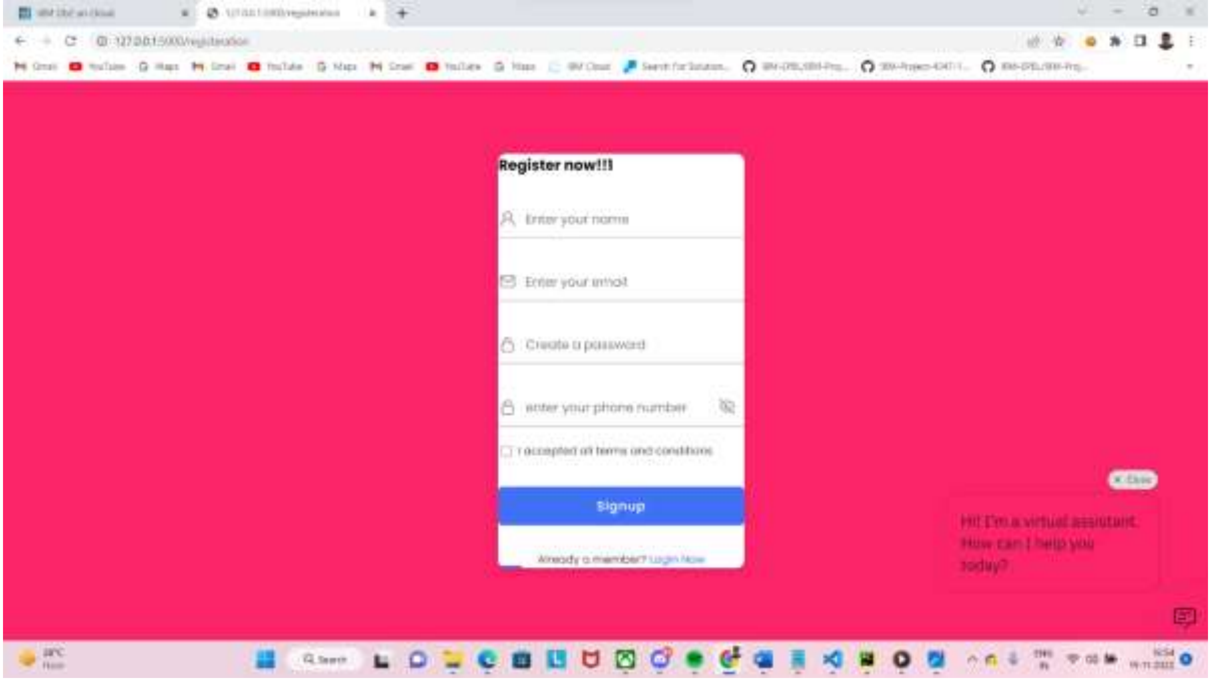
9.1 Performance Metrics



Find the most exciting startup jobs



Hi! I'm a virtual assistant. How can I help you today?



Register now!!!

Enter your name

Enter your email

Create a password

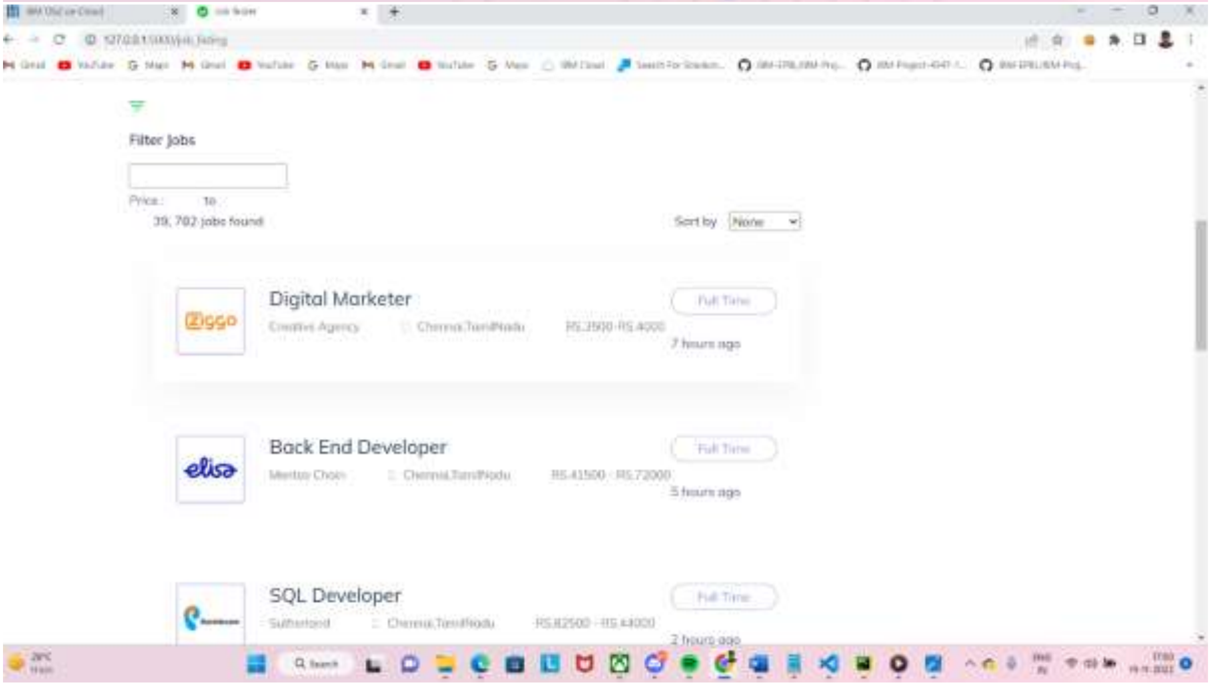
enter your phone number

□ I accepted all terms and conditions

Signup

Already a member? Login Now

Hi! I'm a virtual assistant. How can I help you today?



Filter Jobs

Price : to

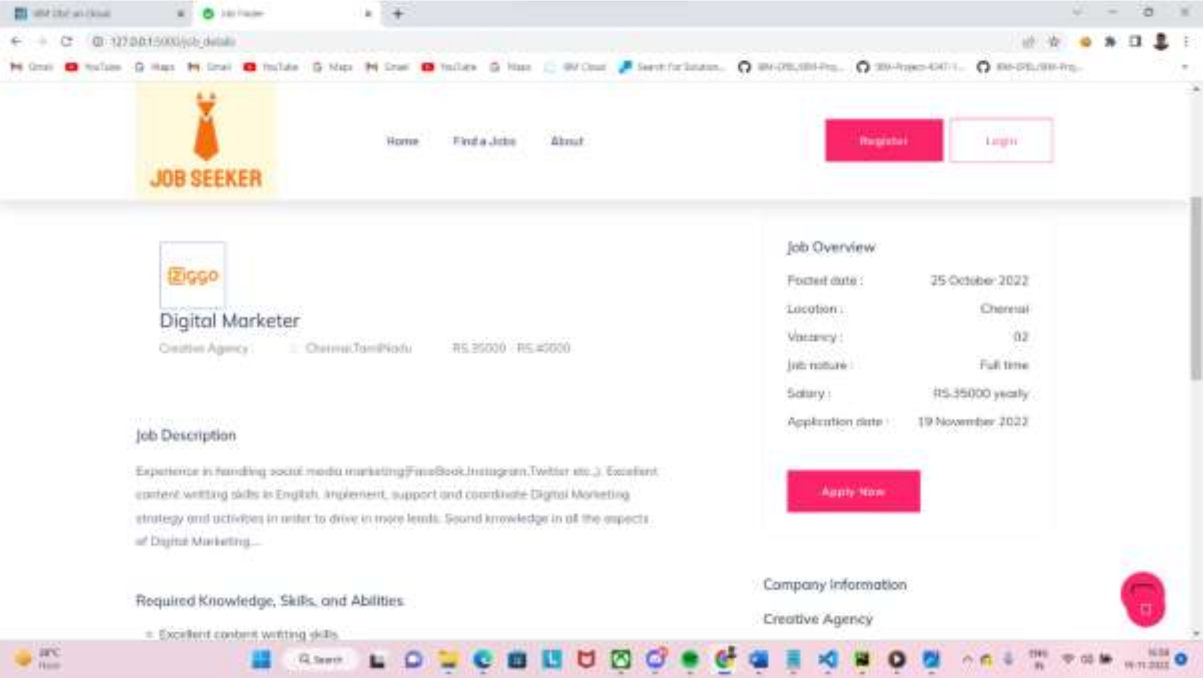
39, 702 jobs found

Sort by None

Digital Marketer
Creative Agency Chennai,TamilNadu RS.3500-RS.4000
Full Time
7 hours ago

Back End Developer
Mentor Chain Chennai,TamilNadu RS.41500 - RS.72000
Full Time
5 hours ago

SQL Developer
Sutherland Chennai,TamilNadu RS.82500 - RS.44000
Full Time
2 hours ago



Digital Marketer

Creative Agency · Chennai,TamilNadu · RS.35000 - RS.40000

Job Overview

Posted date :	25 October 2022
Location :	Chennai
Vacancy :	02
Job nature :	Full time
Salary :	RS.35000 yearly
Application date :	19 November 2022

Job Description

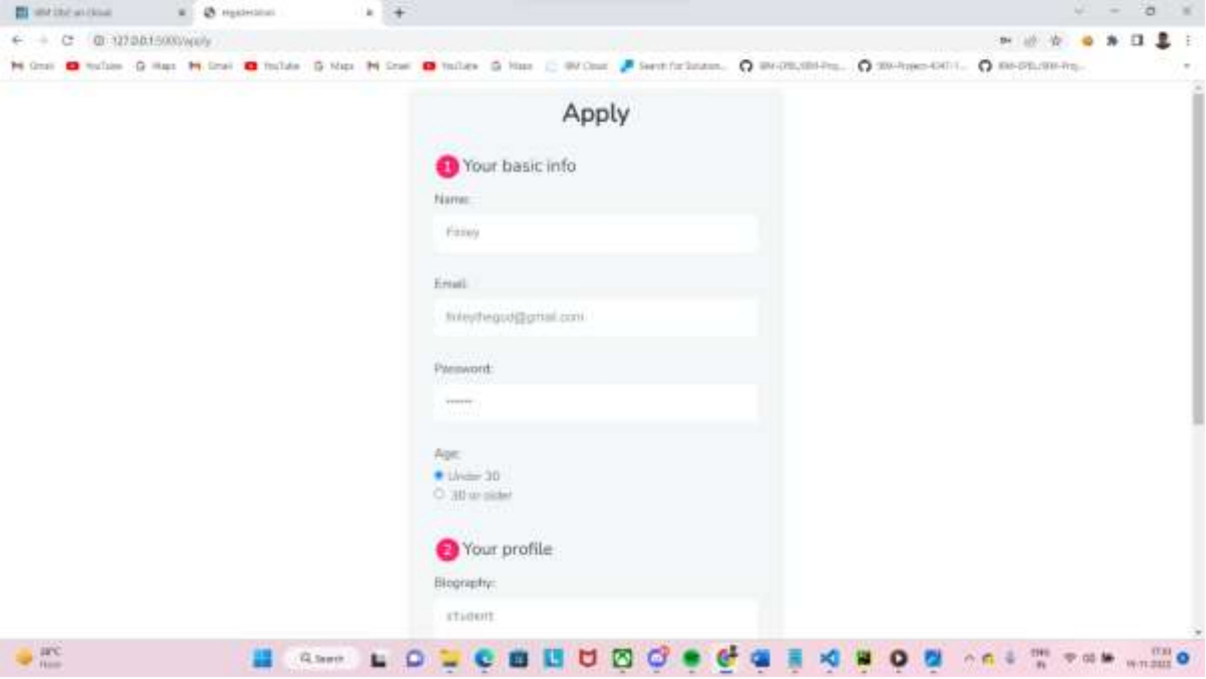
Experience in handling social media marketing(FaceBook,Instagram,Twitter etc.,). Excellent content writting skills in English. Implement, support and coordinate Digital Marketing strategy and activities in order to drive in more leads. Sound knowledge in all the aspects of Digital Marketing....

Required Knowledge, Skills, and Abilities

- Excellent content writting skills

Company Information

Creative Agency



Apply

1 Your basic info

Name:

Finley

Email:

finleythegod@gmail.com

Password:

****

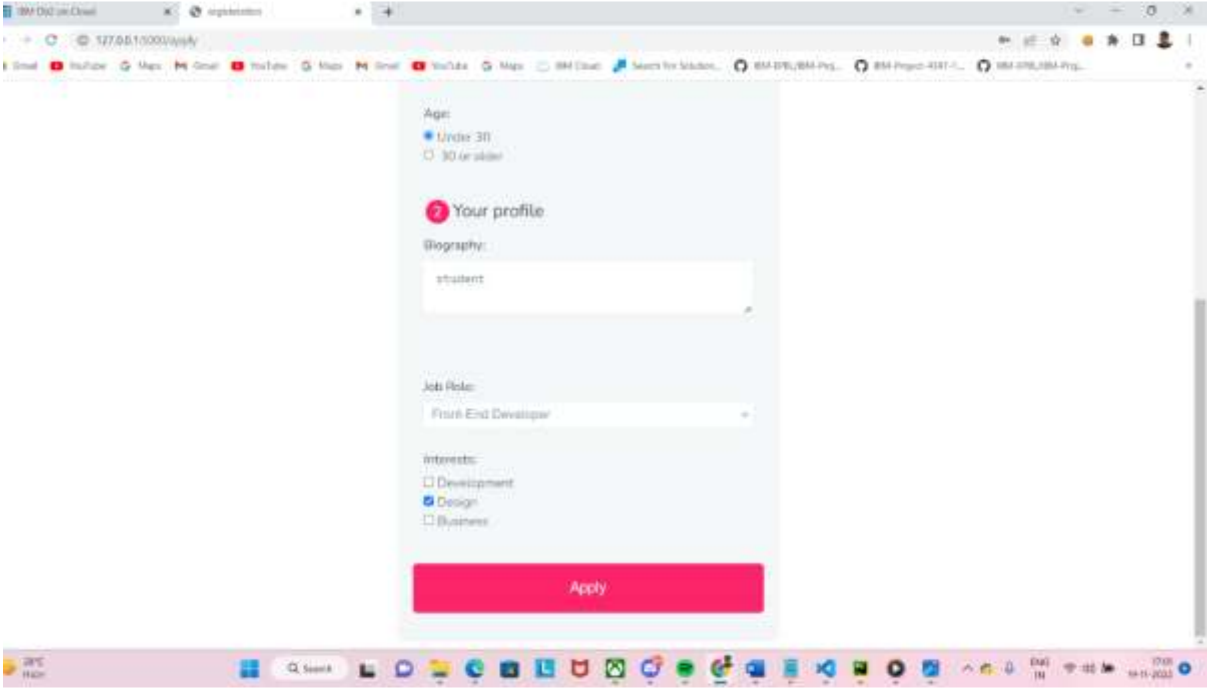
Age:

● Under 30
○ 30 or older

2 Your profile

Biography:

student



127.0.0.1:5000/apply

Age:

● Under 30

○ 30 or older

2 Your profile

Biography:

student

Job Role:

Front End Developer

Interests:

□ Development

☑ Design

□ Business

10. ADVANTAGES & DISADVANTAGES

Advantages:

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

Disadvantages:

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

11. CONCLUSION

By the end of this project, we will

- know fundamental concepts and techniques of recommender system.
- gain a broad understanding of databases and cloud.
- know how to build a web application using the Flask framework.
- know how to build chatbot.

- know how to containerize the application

12. FUTURE SCOPE

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