Skill / Job Recommender Application

Team Id: PNT2022TMID34781

Team Members:

GODWIN.V(962219205023)

JARISH AHAR.J.R(962219205024)

LIVINGSTONE JONES J.E(9622192027)

AKASH S(9622192004)

INDEX

CHAPTER-1 INTRODUCTION

PROJECT OVERVIEW:

In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personalised 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. a) Thus, the contributions of this work are threefold, we 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. b) Put forward the proposal of a framework for job recommendation based on professional skills of job seekers. c) 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 tasks aiming to facilitate research and real-world application design regarding this important issue.

PURPOSE:

The recommender system technology aims to help users in finding items that match their personal interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. Recommender systems help the users to get personalised recommendations,helps users to make correct decisions in their online transactions, increase sales and redefine the users web browsing experience, retain the customers, enhance their shopping experience. Here's where recommendation systems come in handy: they shorten the distance between the customer's need and satisfaction. Not only do they help find the search product; they also discover the needs that customers are not even aware they have. We are primarily used in commercial applications. Examples of such applications include recommending products on Amazon, music on Spotify, and of course, stories on Medium. The famous Netflix Prize is also a competition in the context of recommendation systems. Recommender systems are techniques that serve the best advice for a potential buyer. They suggest the most relevant items to buy and,as a result, increase a company's revenue. These suggestions are based on misbehaviour and history that contain information on their past preferences.

CHAPTER-2

LITERATURE SURVEY REFERENCE

- Adomavicius G, Tuzhilin A (2005). Toward the Next Generation of RecommenderSystems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Trans. Knowl. Data Eng. 17(6):734-749. Barbieri N, Costa G, Manco G, Ortale R (2011).
- Modelling Item Selection andRelevance for Accurate Recommendations: A Bayesian Approach. Proceedings of the fifth ACM conference on Recommender systems (RecSys '11), Chicago, Illinois,USA, ACM pp. 21-28. Keim T (2007).
- Extending the Applicability of Recommender Systems:A MultilayerFramework for Matching Human Resources. In Proceedings of the 40th Annual Hawaii International Conference on System Sci. (HICSS 07). Hawaii, USA, IEEE PROBLEM STATEMENT Problem Statement 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.
- Wecan use a job search API to get the current job openings in the market which will fetch the data directly from the webpage. We have come up with a new innovation solution through which you can directly choose your job related to your skills without needing help from someone.User can login into application and search a job otherwise interact with chatbot via entering skills to the bot, it suggests some job based on entered skills and also it updates latest jobs every day, lists of jobs are uploaded into database.
- The chatbot also connected with the database. Once a user enters skills into a chatbot it will search related jobs in the database then it displays various jobs related to skills.

CHAPTER-3

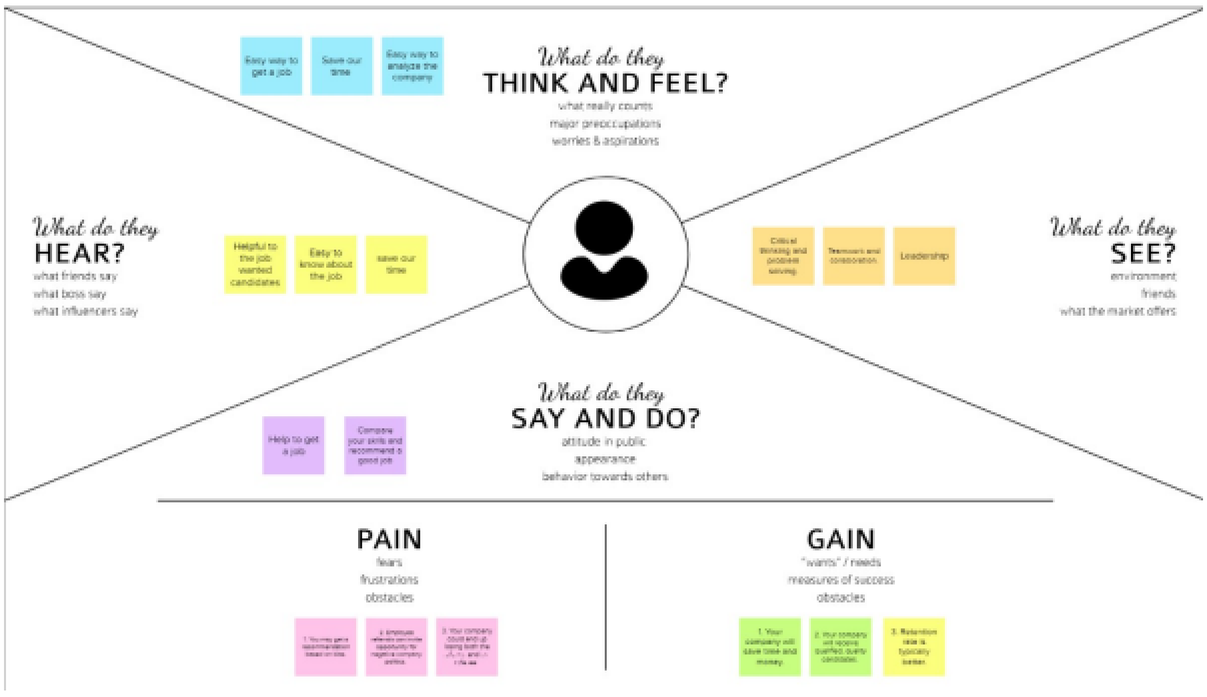
IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS

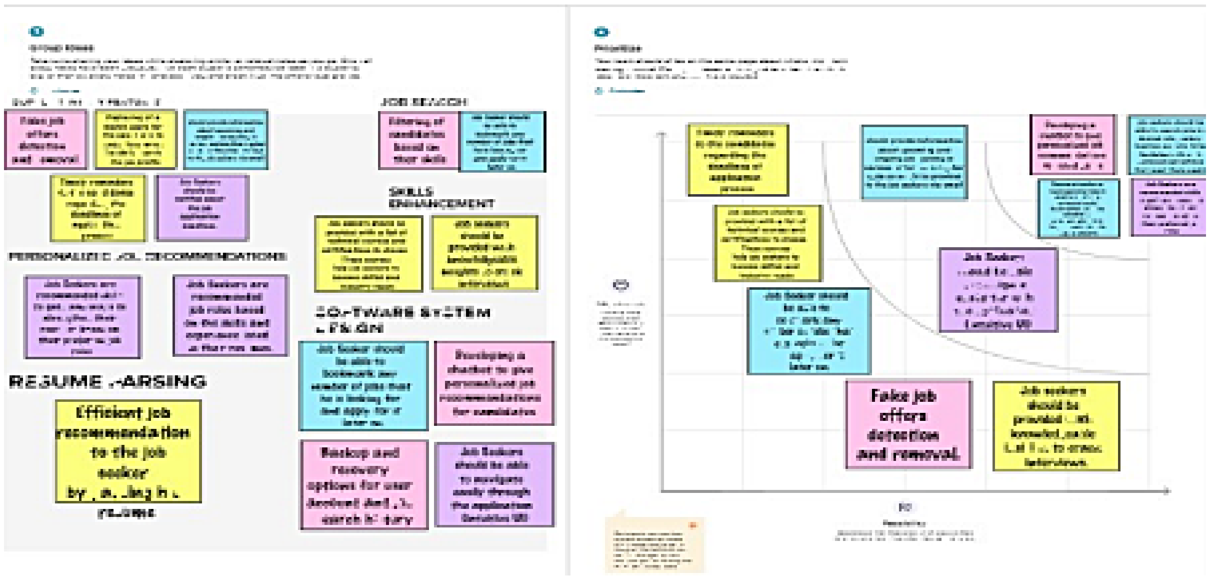
An empathy map is a collaborative tool teams can use to gain a deeper insight into their customers. Much like a user persona, an empathy map can represent a group of users, such as a customer segment.

Objective:

To assess the attitudes towards, applicability and usefulness of empathy maps as part of user's satisfaction in finding a job and connecting them with the industries where they are interested in.



IDEATION AND BRAINSTORMING



Group ideas

JOB SEARCH

Filtering of candidates based on their skills

SKILLS ENHANCEMENT

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.

RESUME PARSING

Efficient job recommendation to the job seeker by parsing his resume

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

Backup and recovery options for user Account and job search history

Job Seekers should be able to navigate easily through the application (Sensitive UI)

Prioritize

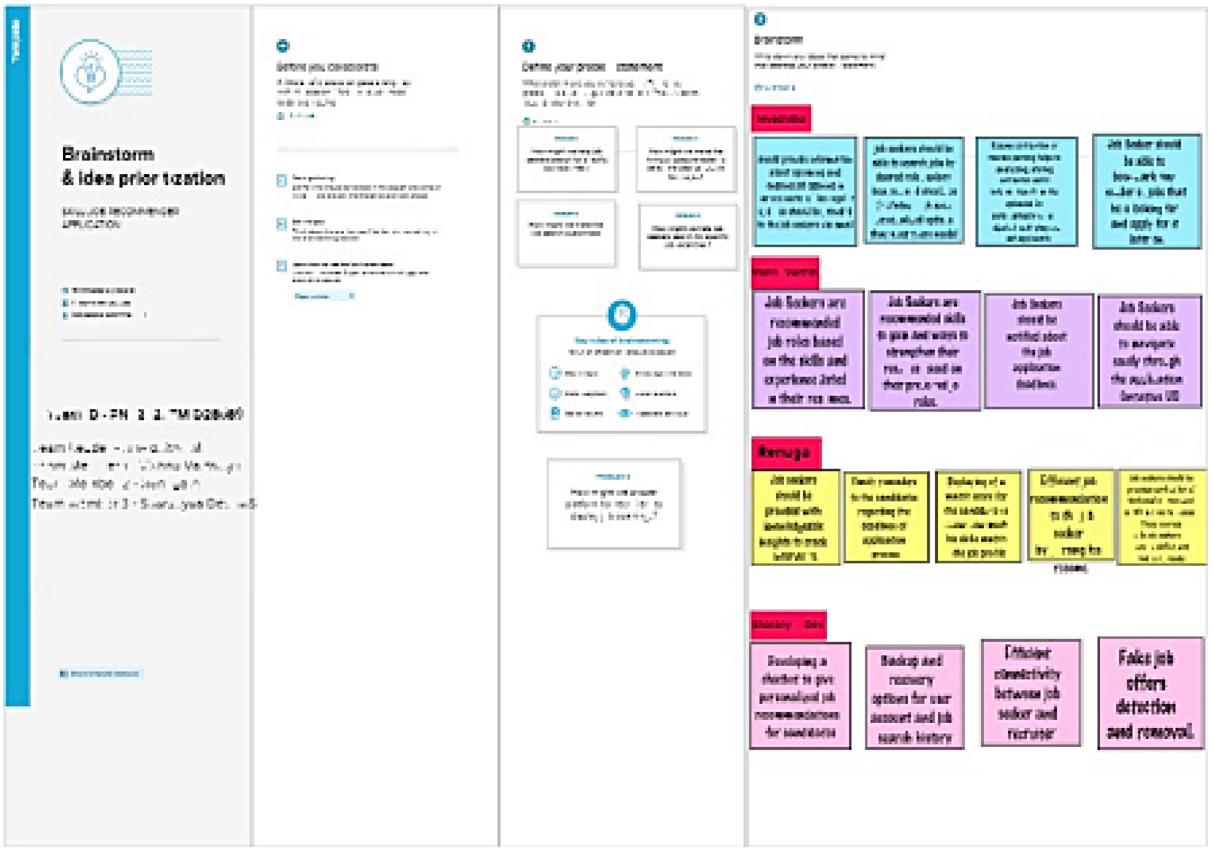
Timely reminders to the candidates regarding the deadlines of application process

Fake job offers detection and removal.

Job seekers should be provided with knowledgeable insights to crack interviews

Importance

Feasibility



Brainstorm & idea prioritization

SKILL/JOB RECOMMENDER APPLICATION

Before you collaborate

Define your problem statement

Brainstorm

Renuga

Job seekers should be provided with knowledgeable insights to crack interviews

Timely reminders to the candidates regarding the deadlines of application process

Efficient job recommendation to the job seeker by parsing his resume

Job Seekers are recommended job roles based on the skills and experience listed in their resumes.

Job Seekers should be notified about the job application deadlines

Job Seekers should be able to navigate easily through the application (Sensitive UI)

Developing a chatbot to give personalized job recommendations for candidates

Backup and recovery options for user account and job search history

Efficient connectivity between job seeker and recruiter

Fake job offers detection and removal.

PROPOSED SOLUTION

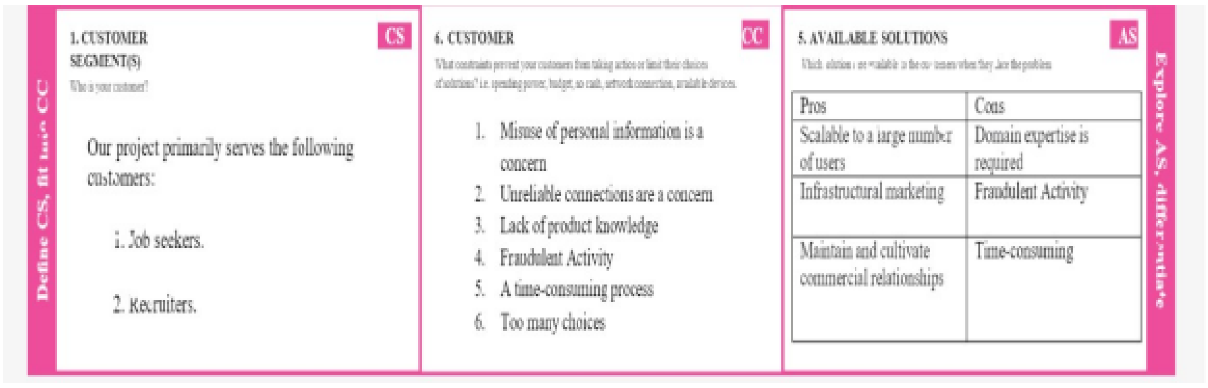
S. No	Parameter	Description
1.	Problem Statement (Problem tobe solved)	Job Recommendation: Recommending the suitable jobs basedon the skills.
2.	Idea / Solution description	We have come up with new innovation solution through which you can directly choose your job related to your skills without need help from someone. You can search a job based on your skills and also chat with chatbot for get recommendation of list of jobs related to specified skillsets. We 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. We can use a job search API toget the current job openings in the market which will fetch the data directly from the webpage. User can login into application and search a job otherwise interact with chatbot via entering skills to the bot, it suggest some job based onentered skills and also it update latest jobs everyday, lists of jobs are uploaded into database and

		the chatbot also connected with database once user enter skills into chatbot it will search related job in database then it display various jobs related to skills.
3.	*Novelty / Uniqueness*	*In this application we provide feature for applying a job that recommended by application and multilingual chatbot for user convinience.*

4.	*Social Impact / Customer Satisfaction*	*Using this application people can easily find their domain based jobs from top MNC's using AI technology that suits their skill set.*
5.	*Business Model (RevenueModel)*	*Revenue can be generated by giving ads in this application and from affiliate commission.*
6.	*Scalability of the Solution*	*Skill and job recommender system will provide job recommendations to any kind of skill set.It never fail to show recommendations.*

PROBLEM SOLUTION FIT

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realised for it actually solves the customer's problem. This system helps the candidates to search for an appropriate job by connecting with the industries and increasing their career growth.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Our project primarily serves the following customers:

1. Job seekers.
2. Recruiters.

6. CUSTOMER CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

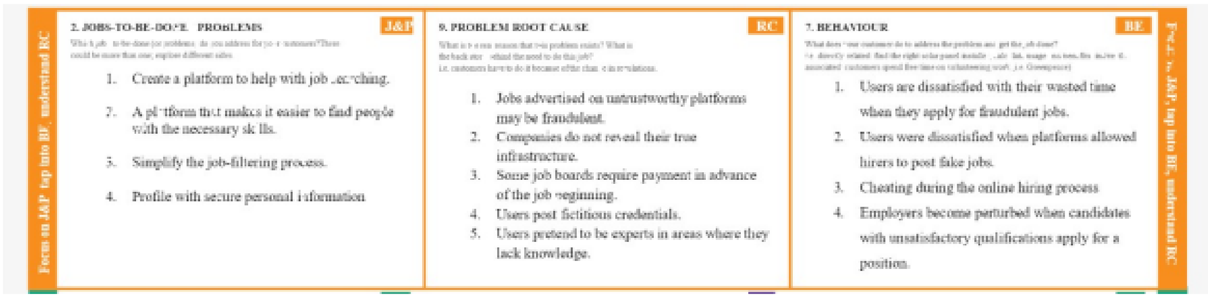
1. Misuse of personal information is a concern
2. Unreliable connections are a concern
3. Lack of product knowledge
4. Fraudulent Activity
5. A time-consuming process
6. Too many choices

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem

Pros	Cons
Scalable to a large number of users	Domain expertise is required
Infrastructural marketing	Fraudulent Activity
Maintain and cultivate commercial relationships	Time-consuming

Explore AS, differentiate



Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1. Create a platform to help with job searching.
2. A platform that makes it easier to find people with the necessary skills.
3. Simplify the job-filtering process.
4. Profile with secure personal information

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists? What is the back story behind the need to do this job? i.e. customers have to do it because of the change in regulations.

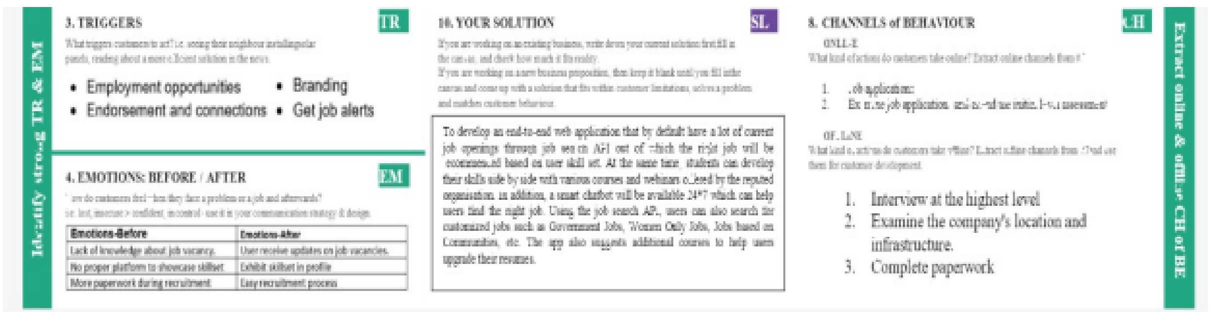
1. Jobs advertised on untrustworthy platforms may be fraudulent.
2. Companies do not reveal their true infrastructure.
3. Some job boards require payment in advance of the job beginning.
4. Users post fictitious credentials.
5. Users pretend to be experts in areas where they lack knowledge.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done? i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

1. Users are dissatisfied with their wasted time when they apply for fraudulent jobs.
2. Users were dissatisfied when platforms allowed hirers to post fake jobs.
3. Cheating during the online hiring process
4. Employers become perturbed when candidates with unsatisfactory qualifications apply for a position.

Focus on J&P, tap into BE, understand RC



Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

- Employment opportunities
- Endorsement and connections
- Branding
- Get job alerts

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards? i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Emotions-Before	Emotions-After
Lack of knowledge about job vacancy.	User receive updates on job vacancies.
No proper platform to showcase skillset.	Exhibit skillset in profile
More paperwork during recruitment	Easy recruitment process

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality. If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

To develop an end-to-end web application that by default have a lot of current job openings through job search API out of which the right job will be recommended based on user skill set. At the same time, students can develop their skills side by side with various courses and webinars offered by the reputed organisation. In addition, a smart chatbot will be available 24*7 which can help users find the right job. Using the job search API, users can also search the customized jobs such as Government Jobs, Women Only Jobs, Jobs based on Communities, etc. The app also suggests additional courses to help users upgrade their resumes.

8. CHANNELS of BEHAVIOUR CH

ONLINE

What kind of actions do customers take online? Extract online channels from #7

1. Job application
2. Examine job application and attend the online assessment

OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

1. Interview at the highest level
2. Examine the company's location and infrastructure.
3. Complete paperwork

Extract online & offline CH of BE

CHAPTER-4 REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Sign in / Login	Register with username, password
FR-2	Profile Registration	Register with username, password, email, qualification, skills. This data will be stored in a database.
FR-3	Job profile display	Display job profiles based on availability, location, skills.
FR-4	Chatbot	A chat on the webpage to solve user queries and issues.
FR-5	Job Registration	The company's registration/Description details will be sent to the registered email id of the user.
FR-6	Logout	Use logout option after completing job registration process.

Non-functional Requirements:

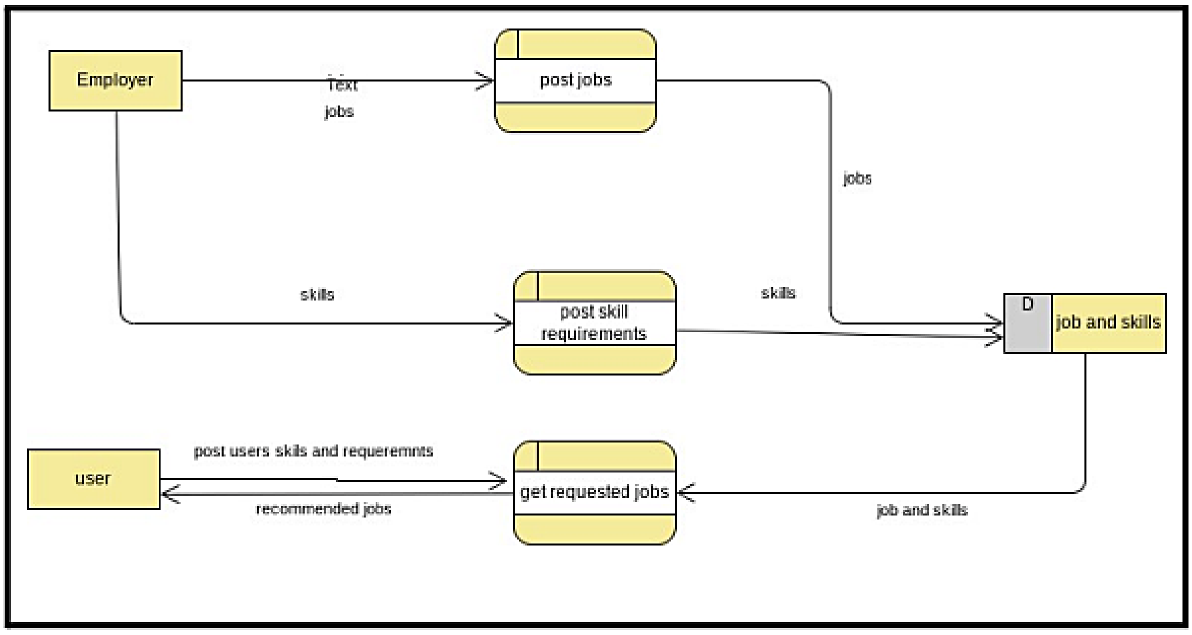
Following are the non-functional requirements of the proposed solution

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The webpage will be designed in such a way that any non-technical user can easily navigate through it and complete the job registration work. (easy and simple design)
NFR-2	**Security**	Using of python flask to cloud connect will provide security to the project. Database will be safely stored in DB2.
NFR-3	**Reliability**	To make sure the webpage doesn't go down due to network traffic.
NFR-4	**Performance**	Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic.
NFR-5	**Availability**	The webpage will be available to all users (network connectivity is necessary) at any given point of time.
NFR-6	**Scalability**	Increasing the storage space of database can increase the number of users. Add some features in future to make the webpage unique and attractive.

CHAPTER-5
PROJECT DESIGN

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.



SOLUTION AND TECHNICAL ARCHITECTURE

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table

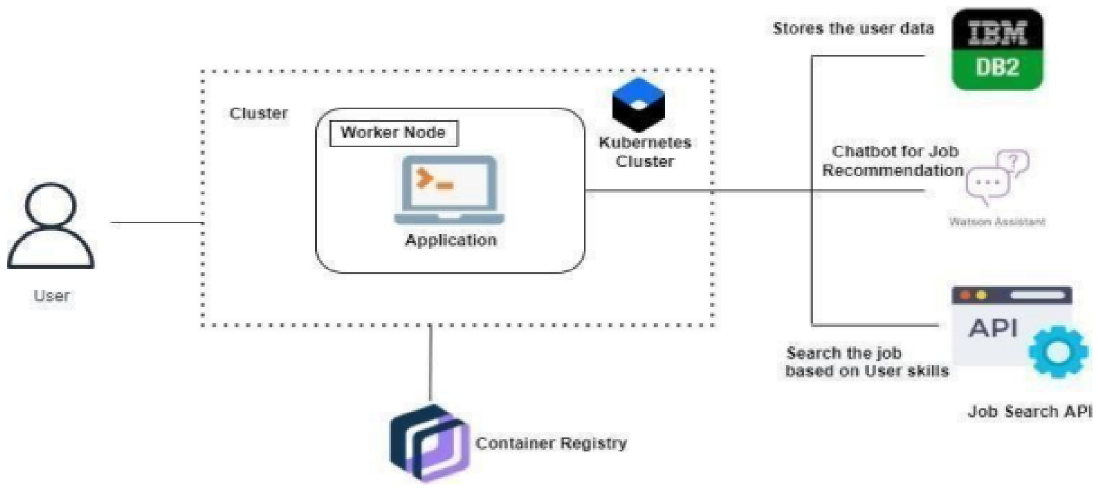


Table-1: Components & Technologies:

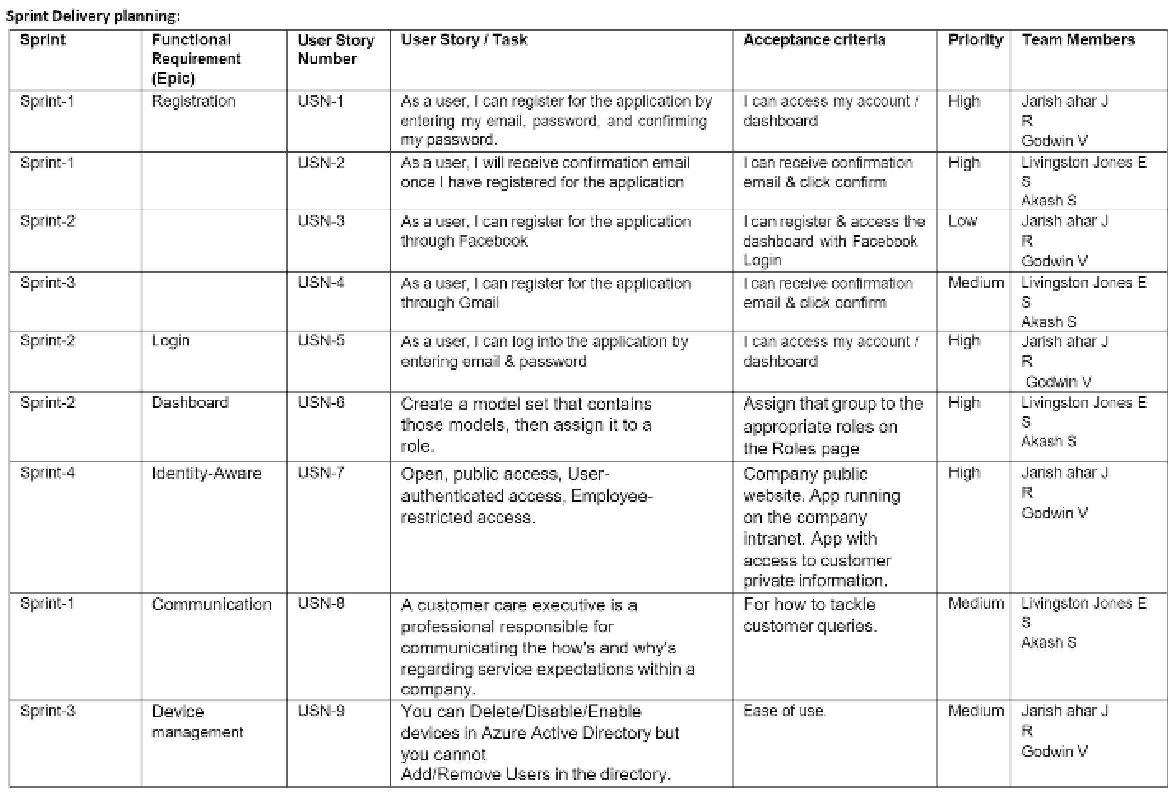
S.No	Component	Description	Technology
1.	User Interface	The user can interacts with our application with the help of chatbot, etc.	HTML, CSS, JavaScript / AngularJs / React Js etc.
2.	Application Logic-1	The User can login with application, by previously he should register in our web app.	Javascript
3.	Application Logic-2	They can also register with the help of a chatbot.	IBM Watson Assistant
4.	Cloud Database	The user data will be stored and retrieved with the help of this database.	IBM DB2, IBM Cloudant etc.
5.	File Storage	The user documents like photos, resumes and much more will be stored in cloud bucket, etc.,	IBM Block Storage or Other Storage Service or Local Filesystem
6.	External API	With the help of API, the user can search the job based on their Skillset.	IBM API, etc.
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description
1.	Is it Scalable?	It follows highly scalable technologies that allows application to handle increase in large user data's workload and perform any operation without any problem.
2.	Is it Modifiable?	It is highly Modifiable and Maintenance requires low cost, compared to other applications.
3.	Is the System Robust?	It does not disturb the performance of the computer by not affecting the operating system. It works in minimal hardware systems.

CHAPTER-6
PROJECT PLANNING & SCHEDULING

Use the below template to create product backlog and sprint schedule



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	Jarish ahar J R Godwin V
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	Livingston Jones E S Akash S
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	Jarish ahar J R Godwin V
Sprint-3		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Livingston Jones E S Akash S
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	Jarish ahar J R Godwin V
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	Livingston Jones E S Akash S
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	Jarish ahar J R Godwin V
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 to tackle customer queries.	Medium	Livingston Jones E S Akash S
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	Jarish ahar J R Godwin V

CHAPTER-7
CODING & SOLUTIONING

FEATURES 1:

1.Creating an HTML file
2.Converted to a python file using flask.
3.Connecting IBM database using the python file.
4.Creating a Docker file

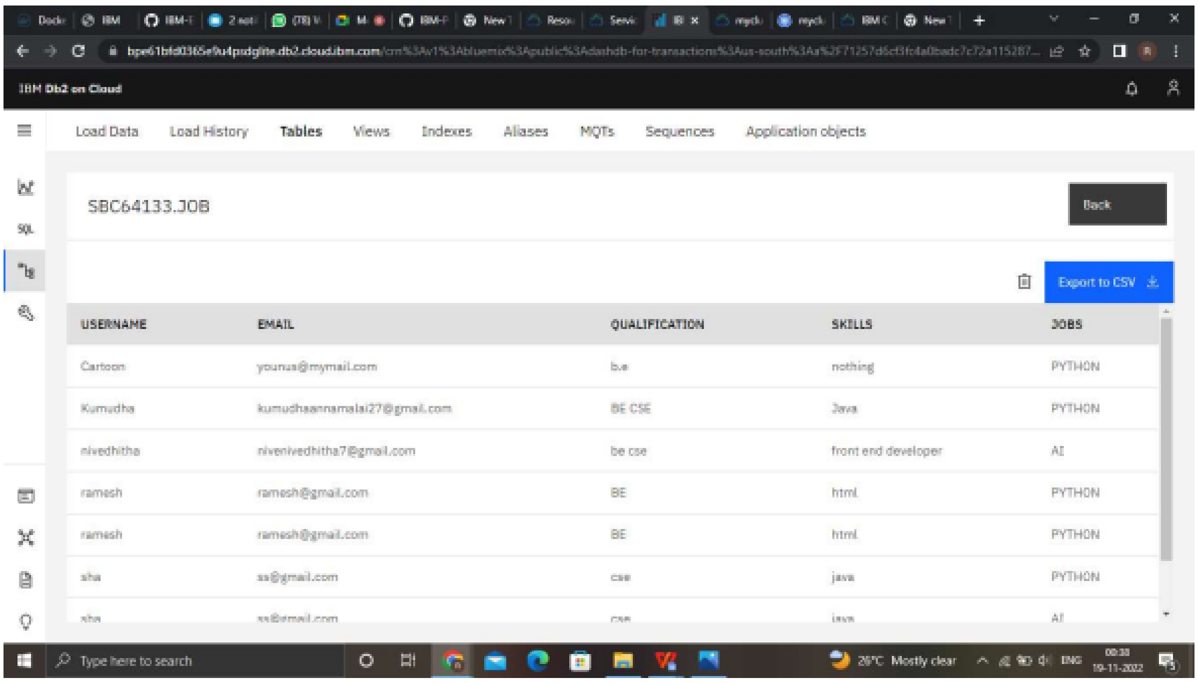
FEATURE

1.Converting the python files to images using Docker Hub.
2.Pushing the images to the container registry.
3.With the help of the Kubernetes Node Port number is obtained
4.Web application can be accessed from anywhere.

DATA SCHEMA

1.UserName
2.Email Id
3.Password
4.Qualifications

5.Skills
6.Domain

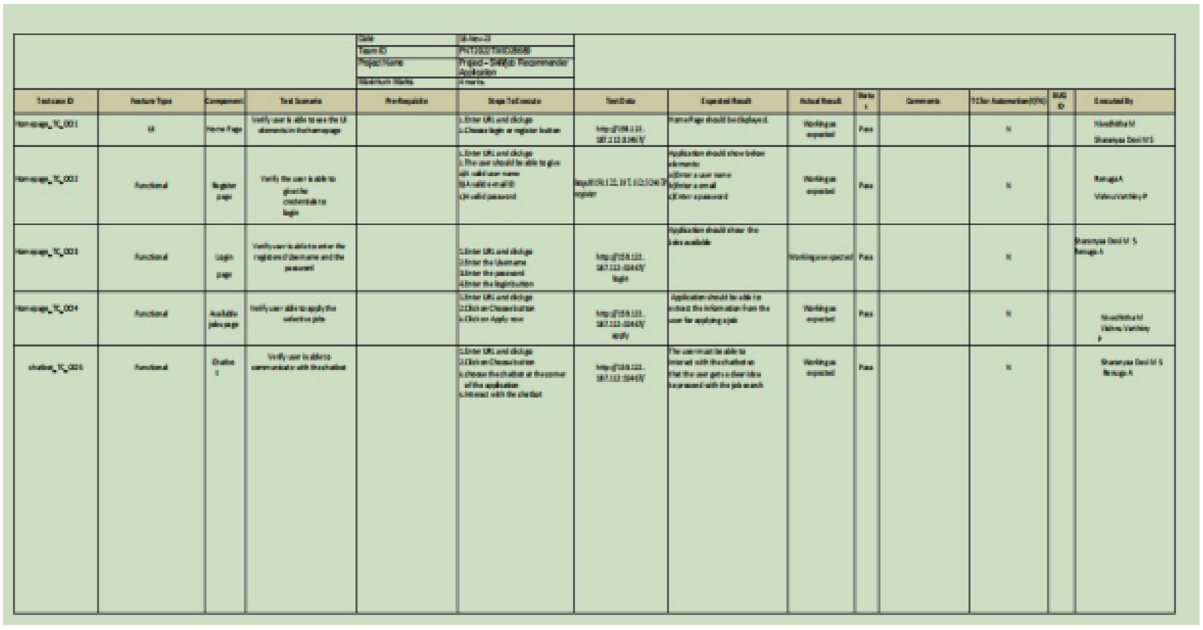


SBC64133.JOB

USERNAME	EMAIL	QUALIFICATION	SKILLS	JOBS
Cartoon	younus@mymail.com	b.e	nothing	PYTHON
Kumudha	kumudhaannamalai27@gmail.com	BE CSE	Java	PYTHON
nivedhitha	nivenivedhitha7@gmail.com	be cse	front end developer	AI
ramesh	ramesh@gmail.com	BE	html	PYTHON
ramesh	ramesh@gmail.com	BE	html	PYTHON
sha	ss@gmail.com	cse	java	PYTHON
sha	ss@gmail.com	cse	java	AI

CHAPTER-8

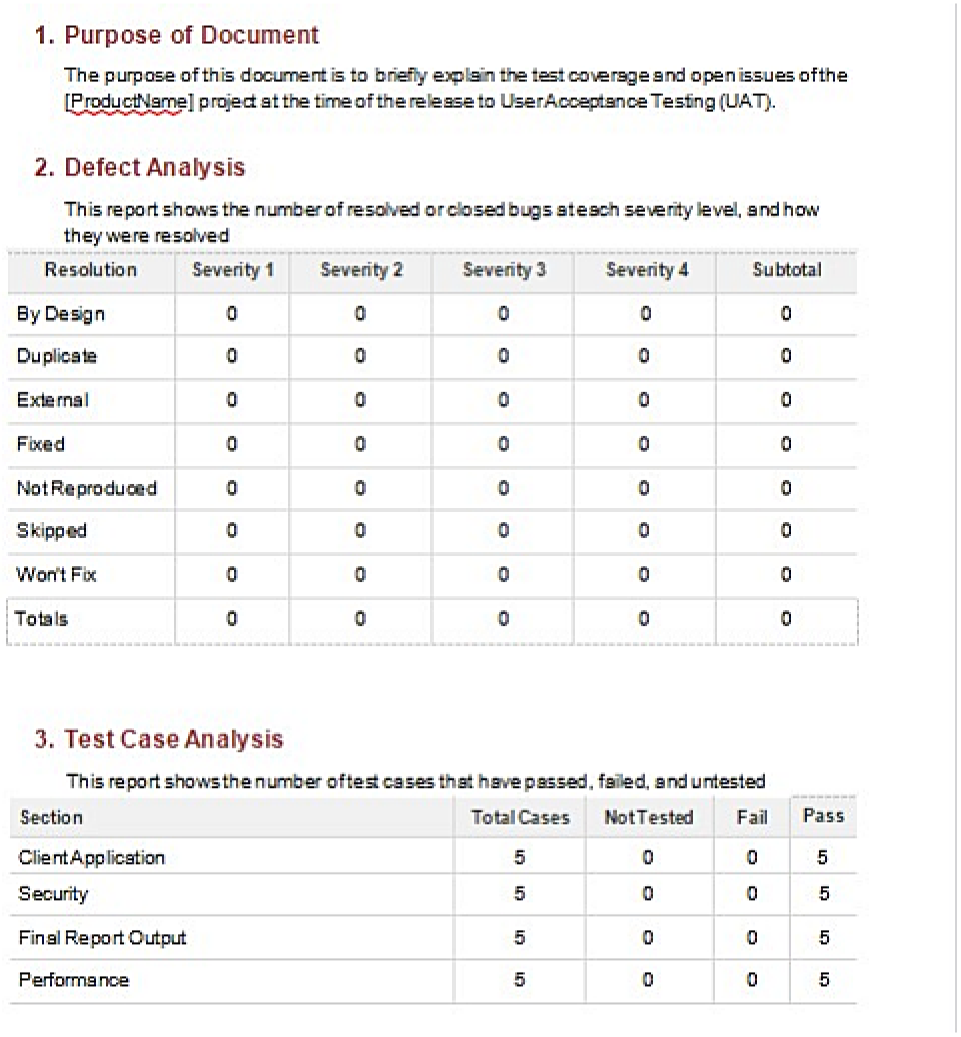
TEST CASES



Date	18-Nov-22
Team ID	PNT2022TMID[illegible]
Project Name	Project – Skill/Job Recommender Application
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Comments	TC for Automation(Y/N)	BUG ID	Executed By
Homepage_TC_001	UI	Home Page	Verify user is able to see the UI elements in the homepage		1.Enter URL and click go 2.Choose login or register button	http://159.122.187.112:30465/	Home Page should be displayed.	Working as expected	Pass		N		Nivedhitha M Sharanya Devi M S
Homepage_TC_002	Functional	Register page	Verify the user is able to give the credentials to login		1.Enter URL and click go 2.The user should be able to give a)A valid user name b)A valid e-mail ID c)A valid password	http://159.122.187.112:30465/register	Application should show below elements: a)Enter a user name b)Enter a email c)Enter a password	Working as expected	Pass		N		Renuga A Vishnu Varthiny P
Homepage_TC_003	Functional	Login page	Verify user is able to enter the registered Username and the password		1.Enter URL and click go 2.Enter the Username 3.Enter the password 4.Enter the login button	http://159.122.187.112:30465/login	Application should show the Jobs available	Working as expected	Pass		N		Sharanya Devi M S Renuga A
Homepage_TC_004	Functional	Available jobs page	Verify user able to apply the selective jobs		1.Enter URL and click go 2.Click on Choose button 3.Click on Apply now	http://159.122.187.112:30465/apply	Application should be able to extract the information from the user for applying a job	Working as expected	Pass		N		Nivedhitha M Vishnu Varthiny P
chatbot_TC_005	Functional	Chatbot	Verify user is able to communicate with the chatbot		1.Enter URL and click go 2.Click on Choose button 3.choose the chatbot at the corner of the application 4.Interact with the chatbot	http://159.122.187.112:30465/	The user must be able to interact with the chatbot so that the user gets a clear idea to proceed with the job search	Working as expected	Pass		N		Sharanya Devi M S Renuga A

USER ACCEPTANCE TESTING



1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] 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

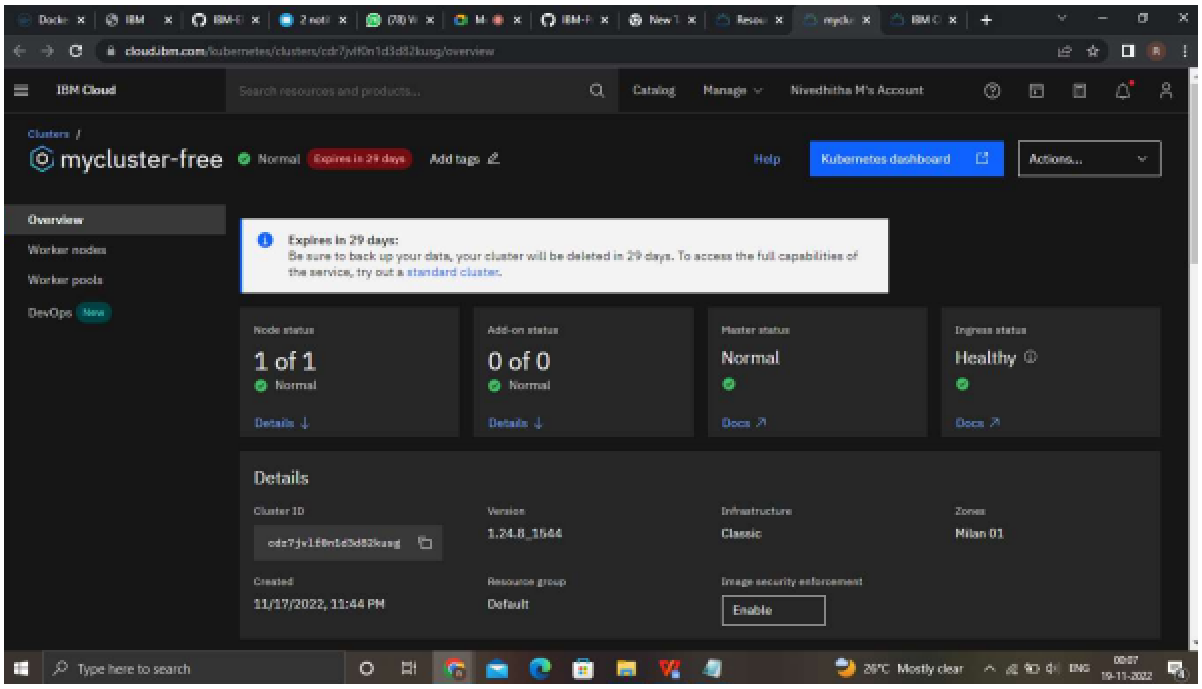
Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	0	0	0	0	0
Duplicate	0	0	0	0	0
External	0	0	0	0	0
Fixed	0	0	0	0	0
Not Reproduced	0	0	0	0	0
Skipped	0	0	0	0	0
Won't Fix	0	0	0	0	0
Totals	0	0	0	0	0

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
Client Application	5	0	0	5
Security	5	0	0	5
Final Report Output	5	0	0	5
Performance	5	0	0	5

CHAPTER-9
PERFORMANCE METRIC



cloud.ibm.com/kubernetes/clusters/cdr7jvlf0n1d3d82kusg/overview

IBM Cloud | Search resources and products... | Catalog | Manage | Nivedhitha M's Account

Clusters /

mycluster-free | Normal | Expires in 29 days | Add tags | Help | Kubernetes dashboard | Actions...

Overview
Worker nodes
Worker pools
DevOps New

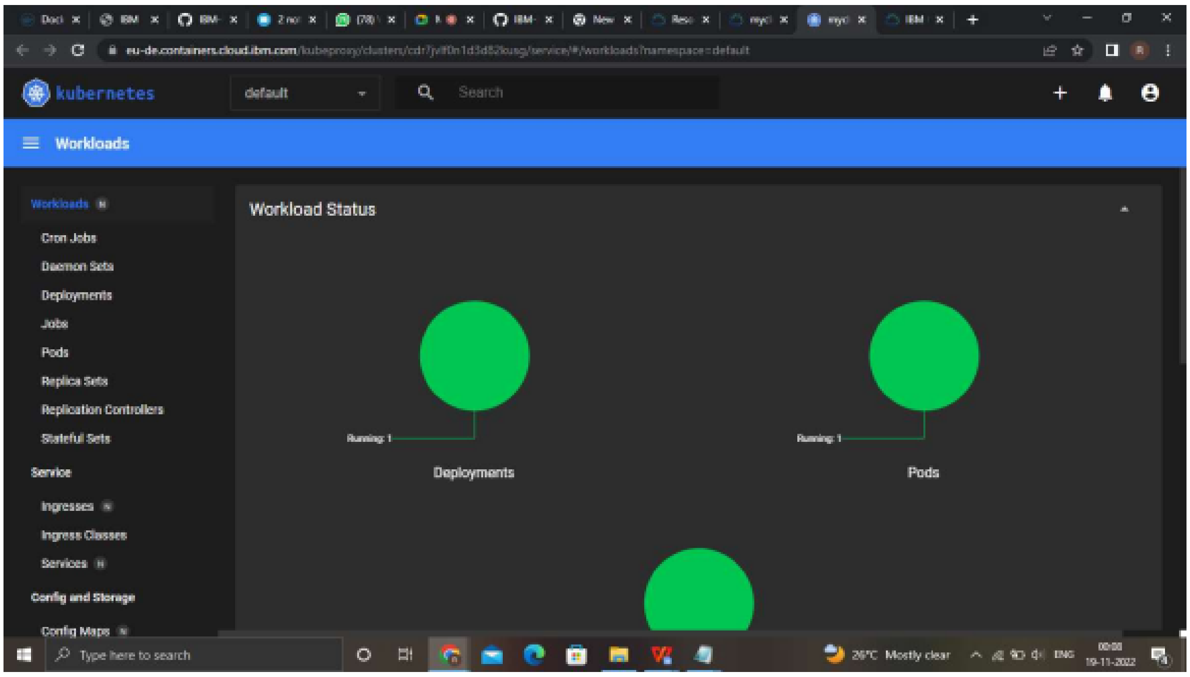
Expires in 29 days:
Be sure to back up your data, your cluster will be deleted in 29 days. To access the full capabilities of the service, try out a standard cluster.

Node status	Add-on status	Master status	Ingress status
1 of 1	0 of 0	Normal	Healthy
Normal	Normal		
Details	Details	Docs	Docs

Details

Cluster ID	Version	Infrastructure	Zones
cdr7jvlf0n1d3d82kusg	1.24.8_1544	Classic	Milan 01
Created	**Resource group**	**Image security enforcement**	
11/17/2022, 11:44 PM	Default	Enable	

Type here to search | 26°C Mostly clear | ENG | 00:07 19-11-2022



eu-de.containers.cloud.ibm.com/kubeproxy/clusters/cdr7jvlf0n1d3d82kusg/service/#/workloads?namespace=default

kubernetes | default | Search

Workloads

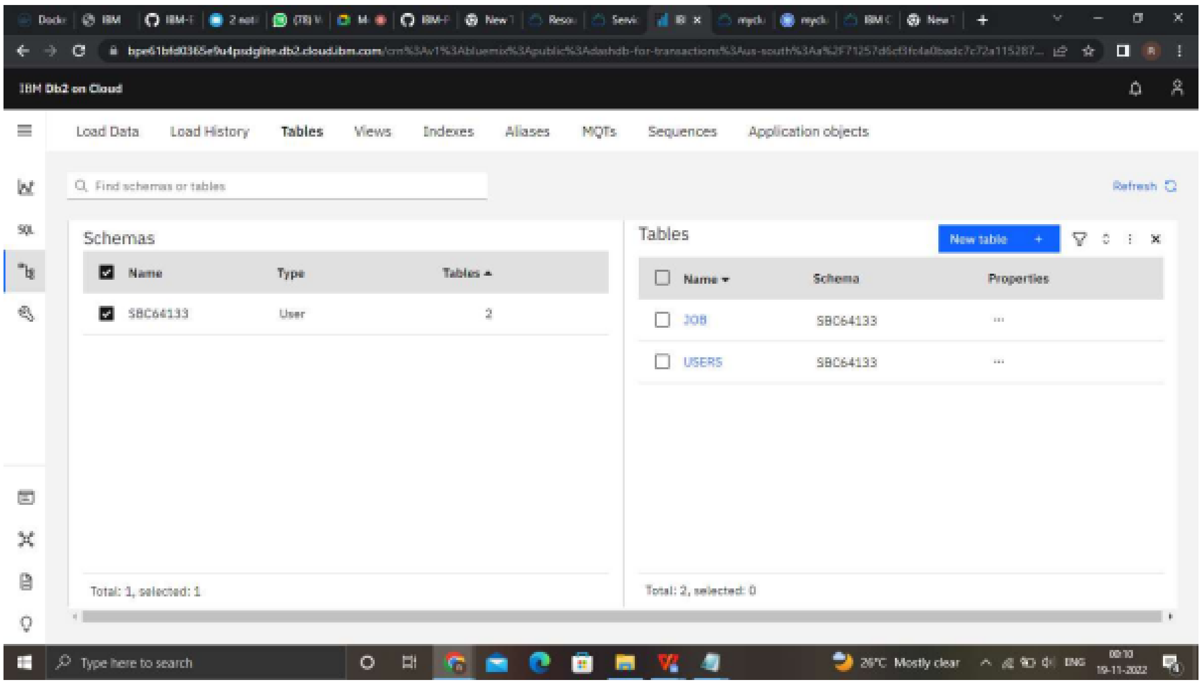
Workloads N
Cron Jobs
Daemon Sets
Deployments
Jobs
Pods
Replica Sets
Replication Controllers
Stateful Sets
Service
Ingresses N
Ingress Classes
Services N
Config and Storage
Config Maps N

Workload Status

Running: 1
Deployments

Running: 1
Pods

Type here to search | 26°C Mostly clear | ENG | 00:08 19-11-2022



Schemas

Name	Type	Tables
SBC64133	User	2

Total: 1, selected: 1

Tables

Name	Schema	Properties
JOB	SBC64133	...
USERS	SBC64133	...

Total: 2, selected: 0

CHAPTER-10
ADVANTAGES AND DISADVANTAGES

advantages

- It can speak volumes for a candidate-in-question when they are referred by an existing employee.
- Not only will the current employee, the referrer, likely want to add to—and not detract from company cultures, but they'll also vouch for required skill sets and competencies.
- Help evaluate your employees' competency profile against those required for their jobs and identify gaps for improvement.
- This data can be used to create personalised development plans that map your employees' paths within the organisation.
- This encourages a substantial interest in specific research fields and technologies .
- The most important fields are Information retrieval and Information filtering.Information retrieval deals with automatically matching user's information

Disadvantages

- The biggest disadvantage of the Job Analysis process is that it is very time consuming.
- It is a major limitation especially when jobs change frequently. If the observer or job analyst is an employee of the same organisation, the process may involve his or her personal likes and dislikes.
- This is a major hindrance in collecting genuine and accurate data. Because of the small sample size, the source of collecting data is extremely small.Therefore, information collected from few individuals needs to be standardised.
- The process involves lots of human efforts. As every job carries different information and there is no set pattern, customised information is to be collected for different jobs.
- The process needs to be conducted separately for collecting and recording job-related data.

CHAPTER-11
CONCLUSION

The system gives a clear vision of the recruiting process and the job recommendation research. We have seen from our literature review and from the challenges that faced the holistic e-recruiting platforms, an increased need for enhancing the quality of candidates/job matching.

The recommender system technologies accomplished significant success in a broad range of applications and potentially a powerful searching and recommending techniques.

Consequently, there is a great opportunity for applying these technologies in the recruitment environment to improve thematching quality. This report shows that several approaches for job recommendation have been proposed, and many techniques combined in order to produce the best fit between jobs and candidates.

We presented state of the art job recommendation as well as a comparative study for its approaches that were proposed by literature. Additionally, we reviewed typical recommender system techniques and the recruiting process related issues.

We conclude that the field of job recommendations is still unripe and require further improvements. As part of our ongoing research, we aim to build a new recommendation approach and test with real data for employee and staffing data from large companies.

CHAPTER-12
FUTURE SCOPE

The tremendous growth of both information and usage has led to a so called information overload problem in which users are finding it increasingly difficult to locate the right information at the right time Thus huge amount of information and easy access to it make recommender systems unavoidable .We use recommender system every day without realising it and without knowing what exactly happens. Recommender systems have changed the way people find products,information, and even other people.

They study patterns of behaviour to know what someone will prefer from among a collection of things he/she has never experienced. Benefits of recommender systems to the businesses using them include: The ability to offer unique personalised service for the customer,Increase trust and customer loyalty, Increase sales, click-through rates,conversions, etc., Opportunities for promotion, persuasion and Obtain more knowledge about customers.

Recommender systems are software tools and techniques providing suggestions for items to be of use to a user. Job recommender systems are desired to attain a high level of accuracy while making the predictions which are relevant to the customer, as it becomes a very tedious task to explore thousands of jobs posted on the web.

APPENDIX

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>
```

manifest.json

```
{
  "short_name": "React App",
  "name": "Create React App Sample",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "64x64 32x32 24x24 16x16",
      "type": "image/x-icon"
    },
    {
      "src": "logo192.png",
      "type": "image/png",
      "sizes": "192x192"
    },
    {
      "src": "logo512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ],
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff"
}
```

navbar.jsx

```
import React, { useContext } from "react";
import { Link, useNavigate } from "react-router-dom";
import { AppContext } from "../context/AppContext";

const Navbar = () => {
  const navigate = useNavigate();
  const { user, setUser, setSkills } = useContext(AppContext);

  const logout = () => {
    setUser(null);
    setSkills([]);
    localStorage.removeItem("user");
    navigate("/");
  };

  return (
    <div className="navbar bg-base-100 border-b-2">
      <div className="flex-1">
        <a
          className="btn btn-ghost normal-case text-xl"
          onClick={() => navigate(user ? "/dashboard" : "/")}
        >
          F-ing Jobs
        </a>
      </div>
      {user && (
        <div className="flex-none gap-2">
          <div className="dropdown dropdown-end">
            <label tabIndex={0} className="btn btn-ghost btn-circle avatar ">
              <div className="w-10 rounded-full ring ring-opacity-50
ring-purple-700">
                <img src="https://placeimg.com/80/80/people" />
              </div>
            </label>
            <ul
              tabIndex={0}
              className="mt-3 p-2 shadow menu menu-compact
dropdown-content bg-base-100 rounded-box w-52"
            >
              <li>
                <a
                  className="justify-between"
                  onClick={() => navigate("/profile")}
                >
                  Profile
                </a>
              </li>
              <li>
                <a onClick={logout}>Logout</a>
              </li>
            </ul>
          </div>
        </div>
      )}
    </div>
  );
};

export default Navbar;
```

appcontext.jsk

```
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);

  const [showAlert, setShowAlert] = 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, showAlert, setShowAlert, skills, setSkills }}
    >
      {children}
    </AppContext.Provider>
  );
};
```

backend_api.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);
  }
};
```

login.jsx

```
import React, { useContext, useState } from "react";
import { Link, useNavigate } from "react-router-dom";
import { AppContext } from "../context/AppContext";
import { loginUser } from "../proxies/backend_api";
import { emailRegex } from "../utils/helper";

const Login = () => {
  const { setShowAlert, setUser } = useContext(AppContext);

  const navigate = useNavigate();

  const [inputs, setInputs] = useState({
```

```
  email: "",
  password: "",
});

const [error, setErrors] = useState({
  email: "",
  password: "",
});

const handleChange = ({ target: { name, value } }) => {
  setErrors((prev) => {
    return { ...prev, [name]: "" };
  });
  setInputs((prev) => ({ ...prev, [name]: value }));
};

const checkInputErrors = () => {
  let status = true;
  if (inputs.email.trim() === "" || !emailRegex.test(inputs.email.trim())) {
    setErrors((prev) => {
      return { ...prev, email: "Enter a valid email" };
    });
    status = false;
  }

  if (inputs.password.trim() === "") {
    setErrors((prev) => {
      return { ...prev, password: "Enter a valid password" };
    });
    status = false;
  }

  if (inputs.password.trim().length < 6) {
    setErrors((prev) => {
      return { ...prev, password: "Minimum 6 characters" };
    });
    status = false;
    }
    return status;
  };

  const handleLogin = async () => {
    if (checkInputErrors()) {
      const data = await loginUser(inputs);
      if (data.error) {
        setShowAlert({ type: "error", message: data.error, duration: 3000 });
        return;
      }
      setUser(data);
      setShowAlert({
        type: "success",
        message: `Welcome back ${data.name}`,
        duration: 3000,
      });
      localStorage.setItem("user", JSON.stringify(data));
      navigate("/dashboard");
    }
  };

  return (
    <div className="flex flex-col justify-center items-center gap-10 mt-5">
      <div>
        <button className="bg-base-300 rounded-box flex flex-row
justify-evenly items-centre gap-10 px-10 py-5 w-fit mx-auto">
          <span>Sign in with Github</span>
          <img src={`github-dark.png`} alt="github" width="14%" />
        </button>
        <div className="divider max-w-xs">or</div>
        <form
          onSubmit={(e) => e.preventDefault()}
          className="card bg-base-300 rounded-box flex flex-col
justify-centre items-centre gap-5 px-10 py-5 w-fit mx-auto"
        >
```

```
<div>
  <input
    value={inputs.email}
    type="text"
    name="email"
    placeholder="email"
    className="input input-bordered input-primary w-full"
    onChange={handleChange}
  />
  {error.email !== "" && (
    <p className="text-sm text-red-500 mt-1 font-medium">
      {error.email}
    </p>
  )}
</div>
<div>
  <input
    value={inputs.password}
    type="password"
    name="password"
    placeholder="password"
    className="input input-bordered input-primary w-full"
    onChange={handleChange}
  />
  {error.password !== "" && (
    <p className="text-sm text-red-500 mt-1 font-medium">
      {error.password}
    </p>
  )}
</div>
<div className="text-center">
  <button
    type="submit"
    onClick={handleLogin}
    className="btn btn-sm btn-primary mb-4"
  >
```

```
          Login
        </button>
        <p>
          Don't have an account?{" "}
          <Link className="text-blue-400" to="/signup">
            Sign up
          </Link>
        </p>
      </div>
    </form>
  </div>
 </div>
 );
};

export default Login;
```

<u>signup.jsk</u>

```
import React, { useContext, useState } from "react";
import { Link, useNavigate } from "react-router-dom";
import { AppContext } from "../context/AppContext";
import { registerUser } from "../proxies/backend_api";
import { emailRegex } from "../utils/helper";

const SignUp = () => {
  const { setShowAlert, setUser } = useContext(AppContext);

  const navigate = useNavigate();

  const [inputs, setInputs] = useState({
    name: "",
    email: "",
    phone_number: "",
    password: "",
    confirm_password: "",
  });
```

```
const [error, setErrors] = useState({
  name: "",
  email: "",
  phone_number: "",
  password: "",
  confirm_password: "",
});

const handleChange = ({ target: { name, value } }) => {
  setErrors((prev) => {
    return { ...prev, [name]: "" };
  });
  setInputs((prev) => ({ ...prev, [name]: value }));
};

const checkInputErrors = () => {
  let status = true;
  if (inputs.email.trim() === "" || !emailRegex.test(inputs.email.trim())) {
    setErrors((prev) => {
      return { ...prev, email: "Enter a valid email" };
    });
    status = false;
  }

  if (inputs.name.trim() === "") {
    setErrors((prev) => {
      return { ...prev, name: "Enter a valid name" };
    });
    status = false;
  }

  if (inputs.phone_number.trim() === "") {
    setErrors((prev) => {
      return { ...prev, phone_number: "Enter a valid phone number" };
    });
    status = false;
  }

  if (inputs.confirm_password.trim() === "") {
    setErrors((prev) => {
      return { ...prev, confirm_password: "Enter a valid  password" };
    });
    status = false;
  }

  if (inputs.password.trim() === "") {
    setErrors((prev) => {
      return { ...prev, password: "Enter a valid password" };
    });
    status = false;
  }

  if (inputs.password.trim().length < 6) {
    setErrors((prev) => {
      return { ...prev, password: "Minimum 6 characters" };
    });
    status = false;
  }

  if (inputs.password.trim() !== inputs.confirm_password.trim()) {
    setErrors((prev) => {
      return { ...prev, confirmPassword: "Password don't match" };
    });
    status = false;
  }
  return status;
};

const handleSignUp = async () => {
  if (checkInputErrors()) {
    const data = await registerUser(inputs);
    if (data.error) {
      setShowAlert({ type: "error", message: data.error, duration: 3000 });
```

```
        return;
      }
      setUser(data);
      setShowAlert({
        type: "success",
        message: `Your journey starts here ${data.name}`,
        duration: 3000,
      });
      localStorage.setItem("user", JSON.stringify(data));
      navigate("/profile");
    }
  };

  return (
    <div className="flex flex-col justify-center items-center gap-10 mt-5">
      <div>
        <button className="bg-base-300 rounded-box flex flex-row
justify-evenly items-center gap-10 px-10 py-5 w-fit mx-auto">
          <span>Sign in with Github</span>
          <img src={`github-dark.png`} alt="github" width="14%" />
        </button>
        <div className="divider max-w-xs">or</div>
        <div className="card bg-base-300 rounded-box flex flex-col
justify-center items-center gap-3 px-10 py-5 w-fit mx-auto">
          <div>
            <input
              value={inputs.name}
              type="text"
              name="name"
              placeholder="name"
              className="input input-bordered input-primary w-full"
              onChange={handleChange}
            />
            {error.name !== "" && (
              <p className="text-sm text-red-500
font-medium">{error.name}</p>
```

```
          )}
        </div>
        <div>
          <input
            value={inputs.email}
            type="text"
            name="email"
            placeholder="email"
            className="input input-bordered input-primary w-full"
            onChange={handleChange}
          />
          {error.email !== "" && (
            <p className="text-sm text-red-500
font-medium">{error.email}</p>
          )}
        </div>
        <div>
          <input
            value={inputs.phone_number}
            type="text"
            name="phone_number"
            placeholder="phone number"
            className="input input-bordered input-primary w-full"
            onChange={handleChange}
          />
          {error.phone_number !== "" && (
            <p className="text-sm text-red-500  font-medium">
              {error.phone_number}
            </p>
          )}
        </div>
        <div>
          <input
            value={inputs.password}
            type="password"
            name="password"
```

```
      placeholder="password"
      className="input input-bordered input-primary w-full"
      onChange={handleChange}
    />
    {error.password !== "" && (
      <p className="text-sm text-red-500  font-medium">
        {error.password}
      </p>
    )}
  </div>
  <div>
    <input
      value={inputs.confirm_password}
      type="password"
      name="confirm_password"
      placeholder="confirm password"
      className="input input-bordered input-primary w-full"
      onChange={handleChange}
    />
    {error.confirm_password !== "" && (
      <p className="text-sm text-red-500  font-medium">
        {error.confirm_password}
      </p>
    )}
  </div>
  <div className="text-center">
    <button
      onClick={handleSignUp}
      className="btn btn-sm btn-primary mb-4"
    >
      Sign Up
    </button>
    <p>
      Already have an account?{" "}
      <Link className="text-blue-400" to="/">
        Sign in
```

```
             </Link>
           </p>
         </div>
       </div>
     </div>
   </div>
  );
};

export default SignUp;
```

helper.js

```
export const emailRegex = /^[\w-.]+@([\w-]+\.)+[\w-]{2,4}$/;

export const urlRegex =

/((([A-Za-z]{3,9}:(?:\/\/)?)(?:[-;:&=\+\$,\w]+@)?[A-Za-z0-9.-]+(:[0-9]+)?|(?:w
ww.|[-;:&=\+\$,\w]+@)[A-Za-z0-9.-]+)((?:\/[\+~%\/.\w-_]*)?\??(?:[-\+=&;%@
.\w_]*)#?(?:[\w]*))?)/;

export const BASE_URL = "http://localhost:5000/api";
```

app.jsx

```
import { useEffect } from "react";
import { BrowserRouter, Route, Routes } from "react-router-dom";
import SignUp from "../src/screens/Signup";
import Alert from "./components/Alert";
import Navbar from "./components/Navbar";
import { AppProvider } from "./context/AppContext";
import Dashboard from "./screens/Dashboard";
import Login from "./screens/Login";
import Profile from "./screens/Profile";

function App() {
  useEffect(() => {
    window.watsonAssistantChatOptions = {
      integrationID: "89571b34-dad5-4ad7-8997-6f7e4db74736",
      region: "au-syd",
      serviceInstanceID: "8d893bea-6198-4677-aca6-2e871cac49db",
      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 (
    <BrowserRouter>
      <AppProvider>
        <Navbar />
        <Alert />
        <Routes>
          <Route path="/" element={<Login />} />
          <Route path="/signup" element={<SignUp />} />
          <Route path="/dashboard" element={<Dashboard />} />
          <Route path="/profile" element={<Profile />} />
        </Routes>
      </AppProvider>
    </BrowserRouter>
  );
}

export default App;
```

index.css

```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;
}

#profile-card {
  background-image: url("data:image/svg+xml,%3csvg
xmlns='http://www.w3.org/2000/svg' version='1.1'
xmlns:xlink='http://www.w3.org/1999/xlink'
xmlns:svgjs='http://svgjs.com/svgjs' width='1440' height='560'
```

preserveAspectRatio='none' viewBox='0 0 1440 560'%3e%3cg
mask='url(%26quot%3b%23SvgjsMask1024%26quot%3b)'
fill='none'%3e%3cpath d='M1140.1901296559531
116.34762677598815L1129.4188620722516-6.7685250727561765
1006.3027102235073 4.0027425109452395 1017.0739778072088
127.11889435968956z' fill='rgba(200%2c 217%2c 241%2c 0.4)'
class='triangle-float2'%3e%3c/path%3e%3cpath d='M298.23 260.8
a157.93 157.93 0 1 0 315.86 0 a157.93 157.93 0 1 0 -315.86 0z'
fill='rgba(200%2c 217%2c 241%2c 0.4)'
class='triangle-float3'%3e%3c/path%3e%3cpath d='M778.59 516.07
a117.52 117.52 0 1 0 235.04 0 a117.52 117.52 0 1 0 -235.04 0z'
fill='rgba(200%2c 217%2c 241%2c 0.4)'
class='triangle-float1'%3e%3c/path%3e%3cpath
d='M468.9084559539883 415.4002443076678L357.01761862323065
435.12961782314227 376.7469921387052 547.0204551539
488.6378294694628 527.2910816384255z' fill='rgba(200%2c 217%2c
241%2c 0.4)' class='triangle-float2'%3e%3c/path%3e%3cpath d='M33.91
344.01 a115.57 115.57 0 1 0 231.14 0 a115.57 115.57 0 1 0 -231.14 0z'
fill='rgba(200%2c 217%2c 241%2c 0.4)'
class='triangle-float1'%3e%3c/path%3e%3cpath d='M-1.96 18.7 a116.99
116.99 0 1 0 233.98 0 a116.99 116.99 0 1 0 -233.98 0z' fill='rgba(200%2c
217%2c 241%2c 0.4)' class='triangle-float2'%3e%3c/path%3e%3cpath
d='M1091.07 104.37 a170.17 170.17 0 1 0 340.34 0 a170.17 170.17 0 1
0 -340.34 0z' fill='rgba(200%2c 217%2c 241%2c 0.4)'
class='triangle-float2'%3e%3c/path%3e%3cpath
d='M694.1712056255251 443.99968549638584L565.7613309753356
394.70775287477215 516.4693983537218 523.1176275249617
644.8792730039114 572.4095601465754z' fill='rgba(200%2c 217%2c
241%2c 0.4)' class='triangle-float2'%3e%3c/path%3e%3cpath
d='M1156.75 235.79 a127.2 127.2 0 1 0 254.4 0 a127.2 127.2 0 1 0
-254.4 0z' fill='rgba(200%2c 217%2c 241%2c 0.4)'
class='triangle-float2'%3e%3c/path%3e%3c/g%3e%3cdefs%3e%3cmask
id='SvgjsMask1024'%3e%3crect width='1440' height='560'
fill='white'%3e%3c/rect%3e%3c/mask%3e%3cstyle%3e %40keyframes
float1 %7b 0%25%7btransform: translate(0%2c 0)%7d
50%25%7btransform: translate(-10px%2c 0)%7d 100%25%7btransform:

```
translate(0%2c 0)%7d %7d .triangle-float1 %7b animation: float1 5s
infinite%3b %7d %40keyframes float2 %7b 0%25%7btransform:
translate(0%2c 0)%7d 50%25%7btransform: translate(-5px%2c -5px)%7d
100%25%7btransform: translate(0%2c 0)%7d %7d .triangle-float2 %7b
animation: float2 4s infinite%3b %7d %40keyframes float3 %7b
0%25%7btransform: translate(0%2c 0)%7d 50%25%7btransform:
translate(0%2c -10px)%7d 100%25%7btransform: translate(0%2c 0)%7d
%7d .triangle-float3 %7b animation: float3 6s infinite%3b %7d
%3c/style%3e%3c/defs%3e%3c/svg%3e");
  background-size: contain;
  background-repeat: repeat-x;
  background-position: centre;
}
```

main.jsx

```
import React from 'react'
import ReactDOM from 'react-dom/client'
import App from './App'
import './index.css'

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
)
```

package.json

```
{
  "name": "frontend",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.16.5",
    "@testing-library/react": "^13.4.0",
    "@testing-library/user-event": "^13.5.0",
    "react": "^18.2.0",
    "react-dom": "^18.2.0",
    "react-scripts": "5.0.1",
    "web-vitals": "^2.1.4",
    "axios": "^1.1.3",
    "daisyui": "^2.33.0",
    "tailwind css": "^3.1.8",
    "postcss": "^8.4.18"
  },
  "scripts": {
    "start": "vite",
    "build": "vite build",
    "preview": "vite preview",
    "server": "cd backend && flask --debug run"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version",
      "last 1 firefox version",
      "last 1 safari version"
    ]
  }
}
```

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-44276-1660723628