SKILL / JOB RECOMMENDER APPLICATION

DOMAIN NAME : CLOUD APPLICATION DEVELOPMENT

TEAM ID : **PNT2022TMID22561**

TEAM MEMBERS :

LIKESH KUMAR BM	TEAM LEADER	113119UG07050
KISHORE K	TEAM MEMBER	113119UG07049
GOWTHAM K	TEAM MEMBER	113119UG07026
PRASANTH V	TEAM MEMBER	113119UG07068

BACHELOR OF TECHNOLOGY
IN
INFORMATION TECHNOLOGY

VEL TECH MULTI TECH DR.RANGARAJAN DR.SAKUNTHALA ENGINEERING COLLEGE

TABLE OF CONTENTS

ABSTRACT

Machine learning is a sub-field of data science that concentrates on designing algorithms that can learn from and make predictions on the data. Presently recommendation frameworks are utilized to take care of the issue of the overwhelming amount of information in every domain and enable the clients to concentrate on information that is significant to their area of interest. One domain where such recommender systems can play a significant role to help college graduates to fulfill their dreams by recommending a job based on their skill set. Currently, there are plenty of websites that provide heaps of information regarding employment opportunities, but this task is extremely tedious for students as they need to go through large amounts of information to find the ideal job. And many students are not aware of which job is suitable for them. Nowadays, the IT fields are in a boom. Many engineering students are learning some technical skills by doing some courses but they don't know which skill is for which job. Simultaneously, existing job recommendation systems only take into consideration the domain in which the user is interested while ignoring their profile and skillset, which can help recommendobs that are tailor-made for the user. This paper examines the user's resume then compares the knowledge of degree, soft skills, hard skills, and the projects he has done and then only the system recommends the jobs for that user. The system not only recommends the jobs but also shows the score of his/her resume for the respective job. Then, the system also recommends skills to improve the scores of theirMachine learning is a subfield of data science that concentrates on designing algorithms that can learn from and make predictions on the data. Presently recommendation frameworks are utilized to take care of the issue of the overwhelming amount of information in every domain and enable the clients to concentrate on information that is significant to their area of interest. One domain where such recommender systems can play a significant role to help college graduates to fulfill their dreams by recommending a job based on their skill set. Currently, there are plenty of websites that provide heaps of information regarding employment opportunities, but this task is extremely tedious for students as they need to go through large amounts of information to find the ideal job. And many students are not aware of which job is suitable for them. Nowadays, the IT fields are in a boom. Many engineering

students are learning some technical skills by doing some courses but they don’t know which skill is for 4 which job. Simultaneously, existing job recommendation systems only take into consideration the domain in which the user is interested while ignoring their profile and skillset, which can help recommend jobs that are tailor-made for the user. This paper examines the user’s resume then compares the knowledge of degree, soft skills, hard skills, and the projects he has done and then only the system recommends the jobs for that user. The system not only recommends the jobs but also shows the score of his/her resume for the respective job. Then, the system also recommends skills to improve the scores of theirMachine learning is a sub-field of data science that concentrates on designing algorithms that can learn from and make predictions on the data. Presently recommendation frameworks are utilized to take care of the issue of the overwhelming amount of information in every domain and enable the clients to concentrate on information that is significant to their area of interest. One domain where such recommender systems can play a significant role to help college graduates to fulfill their dreams by recommending a job based on their skill set. Currently, there are plenty of websites that provide heaps of information regarding employment opportunities, but this task is extremely tedious for students as they need to go through large amounts of information to find the ideal job. And many students are not aware of which job is suitable for them. Nowadays, the IT fields are in a boom. Many engineering students are learning some technical skills by doing some courses but they don’t know which skill is for which job. Simultaneously, existing job recommendation systems only take into consideration the domain in which the user is interested while ignoring their profile and skillset, which can help recommend jobs that are tailor-made for the user. This paper examines the user’s resume then compares the knowledge of degree, soft skills, hard skills, and the projects he has done and then only the system recommends the jobs for that user. The system not only recommends the jobs but also shows the score of his/her resume for the respective job. Then, the system also recommends skills to improve the scores of their

CHAPTER-1

INTRODUCTION

A recent report claims that most college graduates have difficulty in choosing their domain in their job. Many engineers are trying to shift the domain from their field to IT. So, they are doing some courses in online and randomly searching for a job. Nowadays, IT fields are the targets of many students but they don't know which domain is fit for them. To avoid this situation candidates, need a Job recommendation that analyses the skills to recommend a suitable job for the candidate. The solution is to design a system that reads a resume and their skills. The resumes are going through pre-processing to make the design more efficient. For pre-processing top words and porter Stemmer, Porter Stemmer will make every word their root word, and stop words will remove every meaningless word. This makes the system more efficient. Using of-if reflectorized for both resume and job description. Then compare the skills in the resume and description. For comparing, it uses the Cosine Similarity function and finds the scores of the resume for the respective jobs. Now it sorts the list in descending order with respect to their scores. Now, he got a hierarchical order of jobs from top to bottom. So, he can go with the first job or second which the skill he had already. He can be successful in that domain. The System not only shows the job but also recommends the skills to be improved for the job. Because of this, the candidate can train himself/herself for the future purpose and be a more achievable or talented person in his/her domain. For comparing, it uses the Cosine Similarity function and finds the scores of the resume for the respective jobs. Now it sorts the list in descending order with respect to their scores. Now, he got a hierarchical order of jobs from top to bottom. So, he can go with the first job or second which the skill he had already. He can be successful in that domain. The System not only shows the job but also recommends the skills to be improved for the job. Because of this, the candidate can train himself/herself for the future purpose and be a more achievable or talented person in his/her domain.

1.1 PROJECT OVERVIEW:

To find suitable jobs and their scores, this application receives the resume and has a dataset for a job with their description. It will pre-process the resume and job description with the stop words and porter's steamer. Then it reduces into a meaningful bag of words.

Now the application uses a of-id f reflectorized to convert a raw text into a matrix which makes it easy while compare. The main step is comparing the two bag words. For that, it uses the Cosine Similarity function, which is an angle dependent calculation. By using cosine, it has a list of jobs in descending order with respect to scores. The system will move on to the next progress which is finding the skills to be improved by the candidates. The system will take the resume and the skills dataset then compares both and display the skills which are all not in the resume. The major contribution of this work is as follows: The large MNC businesses use the mechanism currently in place for employment recommendations. The method is employed by businesses, not by regular people. If not, they will charge a small subscription fee to check the user's career options. The system functions for the average guy from city to village to modify this predicament. Because the students would look for employment based on their own skills, this approach will reduce unemployment. This company will also grow more quickly, which will result in more job openings.

1.2.PURPOSE:

The dataset used for this research are sourced from Stack overflow survey data which is modeled as the user data for this research. Another dataset was created by web scrapping the Job board Using R programming language to fulfill the road map.

CHAPTER-2
2 LITERATURE SURVEY

Paper 1 : A survey of job recommendersystems

Author : Shaha T. Al-Otaibi1 and Mourad Ykhlef, Riyadh The Internet-based recruiting platforms become a primary recruitment channel in most companies. While such platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the opportunity of recruiting. The recommender system technology aims to help users in finding items that match their personnel interests; it has a successful usage in e-commerce applications to deal with problems related to information overload efficiently. In order to improve the erecruiting functionality, many recommender system approaches have been proposed. This article will present a survey of erecruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching. 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. In addition to, we plan to enhance the similarity measures that suitable for this problem. In this article, we used a literature analysis of many journals and proceedings related to the recruiting process and the job recommendation researches.

Paper 2 : Job Recommendation based on Job Seeker Skills

Author : : Jorge Valverde-Rebaza Ricardo Puma Paul Bustios Nathalia C, Visibilia, Carlos.

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. In this paper, we proposed a framework for job recommendation task. This framework facilitates the understanding of job recommendation process as well as it allows the use of a variety of text processing and recommendation methods according to the preferences of the job recommender system designer. Moreover, we also contribute making publicly available a new dataset containing job seekers profiles and job vacancies.

Paper 3: Job Seekers' Acceptance of Job Recommender Systems
Author : Sven Laumer, Fabian Guble fabin,maier unibamberg.d

Based on UTAUT2 and the importance of trust to explain user behavior in relation to recommender systems, we focus on job recommender systems by developing and validating a job recommender system acceptance model. The results of our empirical, survey-based study with 440 job seekers indicate that beside performance expectancy and habit, trust is among the three most important determinants and it is especially relevant for women, passive job seekers and those without experience in using job recommender systems. The paper extends general trust and recommender system research by revealing three moderators for the trust and intention relationship. It contextualizes the UTAUT2 by incorporating trust as an antecedent of a consumer's intention to use and by revealing three moderating effects for this relationship. Hence, it is the basis for further studies investigating the acceptance of job recommender system, which has rather been neglected by prior research. Our research was motivated to analyze the influence of trust compared to other variables predicting the intention to use job recommender systems. We observe that performance expectance, hedonic motivations, habit, and trust are important predictors of the intention to use job recommender systems. Hence, job seekers who evaluate the performance of job recommender systems positively, who enjoy using it, who trust it, and who have a habit to use online recruiting services in general have a high intention to use job .

Paper 4: Job Recommendation System Using Profile Matching And Web-Crawling
Author : Deepali V Musale 1, Mamta K Nagpure2, Kaumudini S Patil3, Rukhsar F Sayyed, K K Wagh.

The developed system is job recommendation system for campus recruitment which helps college placement office to match company's profiles and student's profiles with higher precision and lower cost. For profile matching, two matching methods are used: semantic matching, tree-based knowledge matching and query matching. These methods are integrated according to representations of attributes of students and companies, and then the profile similarity degree is acquired. Based on profile similarity degree, preference lists of companies and students are generated. Also students can perform keyword based search for job profiles from various job recruitment sites (e.g. Naukari.com,indeed.com). For obtaining data from online recruitment sites system uses web crawling. With loop matching, matching results would be further optimized and provide more effective guidance for recommendation. In this paper, the efforts were put to take into consideration the job preferences of the candidates along with the content based profile matching, providing SMS based recommendation. Also the jobs are recommended from the online website like naukri.com, etc. The first type of recommendation is done through web portal by using keyword based search and second type of recommendation is done through profile matching and sending notification to the students. Thus proper job recommendations are provided to the students.

2.1.EXISTING PROBLEM :

The major contribution of this work is as follows: The large MNC businesses use the mechanism currently in place for employment recommendations. The method is employed by businesses, not by regular people. If not, they will charge a small subscription fee to check the user's career options. The system functions for the average guy from city to village to modify this predicament. Because the students would look for employment based on their own skills, this approach will reduce unemployment. This company will also grow more quickly, which will result in more job openings. The goal of the proposed work is to suggest a job that is ideal for the user. It displays the hierarchical jobs that are best for the user, not just one job. Additionally, it suggests skills for the jobs that were suggested for the user. This project is intended for someone who simply has no idea what they are going to do. Additionally, there are no logins available because doing so increases the likelihood that users would reject you. The subsequent chapter goes over the specifics of the implementation. The rest of the paper organizes as follows: Chapter 2 provides the literature review conducted for this project. Chapter 3 presents the System Design and Architecture of the project along with the methodology. Chapter 4 discusses the algorithms proposed in this project. Chapter 5 presents the project conclusion and future works on this project

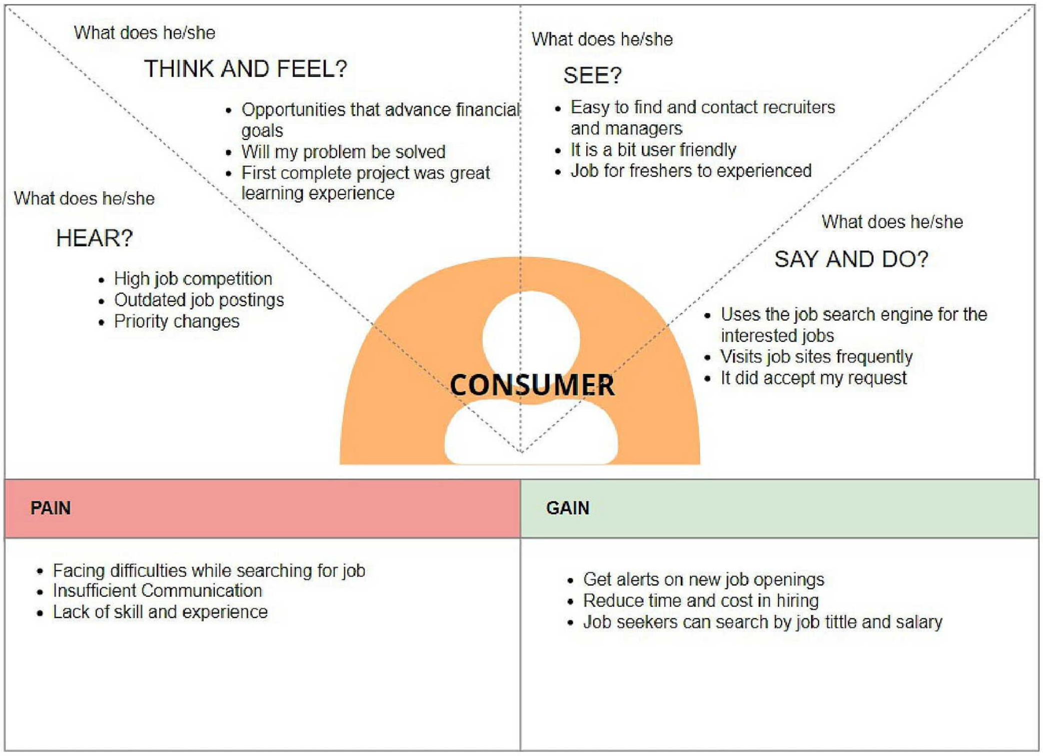
2.2 PROBLEM STATEMENT :

In the last few 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 recommend job vacancies that fit properly to the jobseekers profiles. Thus, the contributions of this work are threefold, made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites, put forward the proposal of a framework for job recommendation based on professional skills of job seekers, and carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. Thus present a general panorama of job recommendation tasks aiming to facilitate research and real-world application design regarding this important issue. Job matching, job seeking, job search, job recommender systems. Proposed a framework for job recommendation tasks. This framework facilitates the understanding of the job recommendation process as well as it allows the use of a variety of text processing and recommendation methods according to the preferences of the job recommender system designer. Moreover, we also contribute to making publicly available a new dataset containing job seekers profiles and job vacancies. Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation.

CHAPTER-3

IDEATION & PROPOSED SOLUTION

3.1. EMPATHY MAP:



What does he/she

THINK AND FEEL?

- Opportunities that advance financial goals
- Will my problem be solved
- First complete project was great learning experience

What does he/she

SEE?

- Easy to find and contact recruiters and managers
- It is a bit user friendly
- Job for freshers to experienced

What does he/she

HEAR?

- High job competition
- Outdated job postings
- Priority changes

What does he/she

SAY AND DO?

- Uses the job search engine for the interested jobs
- Visits job sites frequently
- It did accept my request

CONSUMER

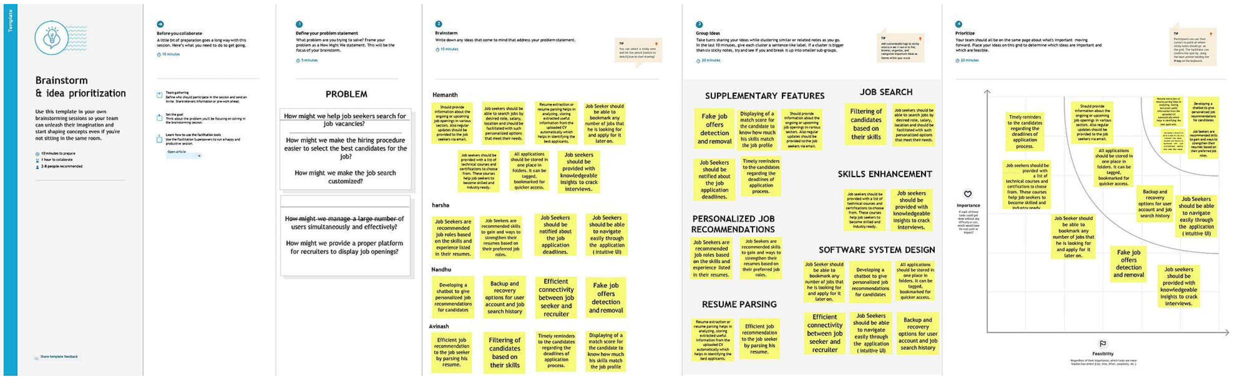
PAIN

- Facing difficulties while searching for job
- Insufficient Communication
- Lack of skill and experience

GAIN

- Get alerts on new job openings
- Reduce time and cost in hiring
- Job seekers can search by job tittle and salary

3.2.Ideation & Brain Storming:



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

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

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

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

Open article

1 Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we help job seekers search for job vacancies?

How might we make the hiring procedure easier to select the best candidates for the job?

How might we make the job search customized?

~~How might we manage a large number~~ of users simultaneously and effectively?

How might we provide a proper platform for recruiters to display job openings?

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!

Hemanth

- Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
- Job seekers should be able to search jobs by desired role, salary, location and should be facilitated with such personalized options that meet their needs.
- Resume extraction or resume parsing helps in analyzing, storing extracted useful information from the uploaded CV automatically which helps in identifying the best applicants.
- Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
- Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
- All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
- Job seekers should be provided with knowledgeable insights to crack interviews.

harsha

- Job Seekers are recommended job roles based on the skills and experience listed in their resumes.
- Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.
- Job Seekers should be notified about the job application deadlines.
- Job Seekers should be able to navigate easily through the application (intuitive UI)

Nandhu

- 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

Avinash

- Efficient job recommendation to the job seeker by parsing his resume.
- Filtering of candidates based on their skills
- Timely reminders to the candidates regarding the deadlines of application process.
- Displaying of a match score for the candidate to know how much his skills match the job profile

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

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

SUPPLEMENTARY FEATURES

- Fake job offers detection and removal
- Displaying of a match score for the candidate to know how much his skills match the job profile
- Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
- Job Seekers should be notified about the job application deadlines.
- Timely reminders to the candidates regarding the deadlines of application process.

JOB SEARCH

- Filtering of candidates based on their skills
- Job seekers should be able to search jobs by desired role, salary, location and should be facilitated with such personalized options that meet their needs.

SKILLS ENHANCEMENT

- Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
- Job seekers should be provided with knowledgeable insights to crack interviews.

PERSONALIZED JOB RECOMMENDATIONS

- Job Seekers are recommended job roles based on the skills and experience listed in their resumes.
- Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.

SOFTWARE SYSTEM DESIGN

- Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
- Developing a chatbot to give personalized job recommendations for candidates
- All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
- Efficient connectivity between job seeker and recruiter
- Job Seekers should be able to navigate easily through the application (intuitive UI)
- Backup and recovery options for user account and job search history

RESUME PARSING

- Resume extraction or resume parsing helps in analyzing, storing extracted useful information from the uploaded CV automatically which helps in identifying the best applicants.
- Efficient job recommendation to the job seeker by parsing his resume.

4 Prioritize

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

20 minutes

TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
Timely reminders to the candidates regarding the deadlines of application process.
Job seekers should be provided with a list of technical courses and certifications to choose from. These courses help job seekers to become skilled and industry ready.
Should provide information about the ongoing or upcoming job openings in various sectors. Also regular updates should be provided to the job seekers via email.
All applications should be stored in one place in folders. It can be tagged, bookmarked for quicker access.
Job Seeker should be able to bookmark any number of jobs that he is looking for and apply for it later on.
Resume extraction or resume parsing helps in analyzing, storing extracted useful information from the uploaded CV automatically which helps in identifying the best applicants.
Developing a chatbot to give personalized job recommendations for candidates
Job Seekers are recommended skills to gain and ways to strengthen their resumes based on their preferred job roles.
Backup and recovery options for user account and job search history
Job Seekers should be able to navigate easily through the application (intuitive UI)
Fake job offers detection and removal
Job seekers should be provided with knowledgeable insights to crack interviews.

3.3.PROPOSED SOLUTION:

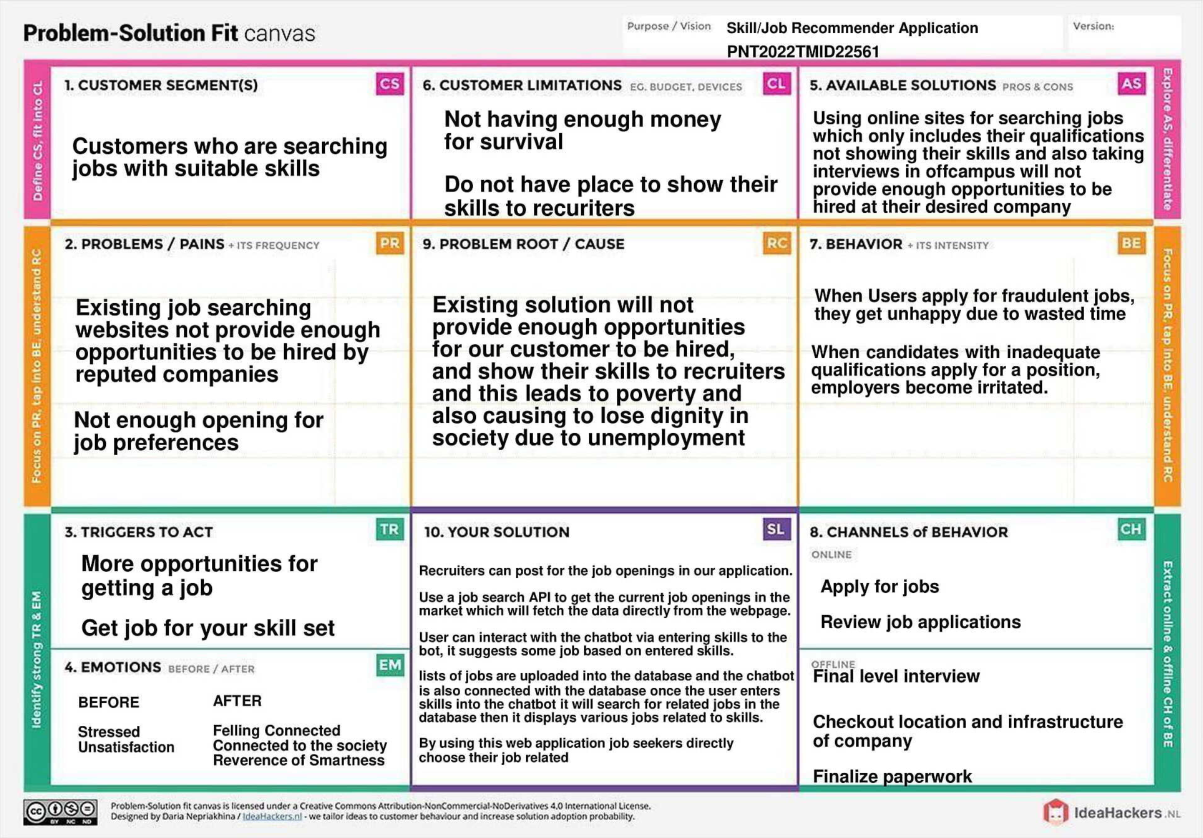
S. No	Parameter	Description
1	Problem Statement (Problem to be solved)	Having lots of skills but wondering which job will best suit you? Don’t need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search 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
2	Idea / Solution description	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 iii) carried out an evaluation to quantify recommendation abilities of two state-ofthe art methods, considering different configurations, within the proposed framework. We thus present a general 13 panorama of job recommendation task aiming to facilitate research and real world application design regarding this important issue
3	Novelty / Uniqueness	The best position are suggested to any person according to her skills. While the position of known profiles are assumed should be noted that there are usually multiple advisable positions corresponding to a set of skills.A recommendation system should return a set of most likely positions and all of them can be equally valid. The recommendation method we use is simply based on representing both positions and

		profiles as comparable vectors and seeking for each profile the positions with the most similar vectors. 4 Social Impact / Customer Satisfaction Students will be benefited as they will get to know which job suits them based on their skill set and therefore Lack of Unemployment can be reduced. 5 Business Model (Revenue Model) We can provide the application for job seekers in a subscription based and we can share the profiles with companies and generate the revenue by providing them best profiles.
4	Social Impact / Customer Satisfaction	
5	Business Model (Revenue Model)	
6	Scalability of the Solution	Data can be scaled up and scaled down according to number of current job openings available

3.4. PROBLEM SOLUTION FIT:

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer’s problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns and recognize what would work and why Purpose:

❑ Solve complex problems in a way that fits the state of your customers.
❑ Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
❑ Sharpen your communication and marketing strategy with the right triggers and messaging
❑ Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.
❑ Understand the existing situation in order to improve it for your target group



Problem-Solution Fit canvas

Purpose / Vision: Skill/Job Recommender Application PNT2022TMID22561

Version:

Define CS, fit into CL

1. CUSTOMER SEGMENT(S) CS

Customers who are searching jobs with suitable skills

6. CUSTOMER LIMITATIONS EG. BUDGET, DEVICES CL

Not having enough money for survival

Do not have place to show their skills to recuriters

5. AVAILABLE SOLUTIONS PROS & CONS AS

Using online sites for searching jobs which only includes their qualifications not showing their skills and also taking interviews in offcampus will not provide enough opportunities to be hired at their desired company

Explore AS, differentiate

Focus on PR, tap into BE, understand RC

2. PROBLEMS / PAINS + ITS FREQUENCY PR

Existing job searching websites not provide enough opportunities to be hired by reputed companies

Not enough opening for job preferences

9. PROBLEM ROOT / CAUSE RC

Existing solution will not provide enough opportunities for our customer to be hired, and show their skills to recruiters and this leads to poverty and also causing to lose dignity in society due to unemployment

7. BEHAVIOR + ITS INTENSITY BE

When Users apply for fraudulent jobs, they get unhappy due to wasted time

When candidates with inadequate qualifications apply for a position, employers become irritated.

Focus on PR, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TO ACT TR

More opportunities for getting a job

Get job for your skill set

4. EMOTIONS BEFORE / AFTER EM

BEFORE	AFTER
Stressed Unsatisfaction	Felling Connected Connected to the society Reverence of Smartness

10. YOUR SOLUTION SL

Recruiters can post for the job openings in our application.

Use a job search API to get the current job openings in the market which will fetch the data directly from the webpage.

User can interact with the chatbot via entering skills to the bot, it suggests some job based on entered skills.

lists of jobs are uploaded into the database and the chatbot is also connected with the database once the user enters skills into the chatbot it will search for related jobs in the database then it displays various jobs related to skills.

By using this web application job seekers directly choose their job related

8. CHANNELS of BEHAVIOR CH

ONLINE

Apply for jobs

Review job applications

OFFLINE

Final level interview

Checkout location and infrastructure of company

Finalize paperwork

Extract online & offline CH of BE

IdeaHackers.NL

CHAPTER-4

REQUIREMENT ANALAYSIS:

4.1 FUNCTIONAL REQUIREMENTS:

S. No	FUNCTIONAL REQUIREMENT (Epic)	SUB REQUIREMENT (Story)
1	Sign In / Login	Register with username, password
2	Profile Registration	Register with username, password, email, qualification, skills. This data will be stored in a database.
3	Job profile display	Display job profiles based on availability, location ,skills
4	Chatbot	A chat on the webpage to solve user queries and issues
5	Job registration	A copy of the company the user applied for with its registration/description details will be sent to the registered email id
6	Logout	

4.2.NON-FUNCTIONAL REQUIREMENTS:

S. No	NON-FUNTIONAL REQUIREMENT	DESCRIPTION
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.)
2	Security	Using of SSL certificate will provide security to the project. Database will be safely stored in DB2.
3	Reliability	To make sure the webpage doesn't go down due to network traffic.
4	Availability	This webpage will be available to all users

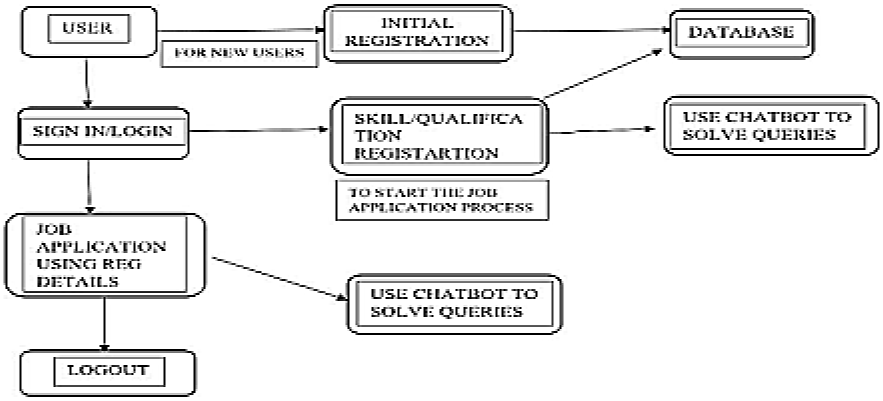
		(network connectivity is necessary) at any given point of time
5	Scalability	Increasing the storage space of database can increase the number of users. Add some features in future to make the webpage unique and attractive
6	Performance	Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic

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.

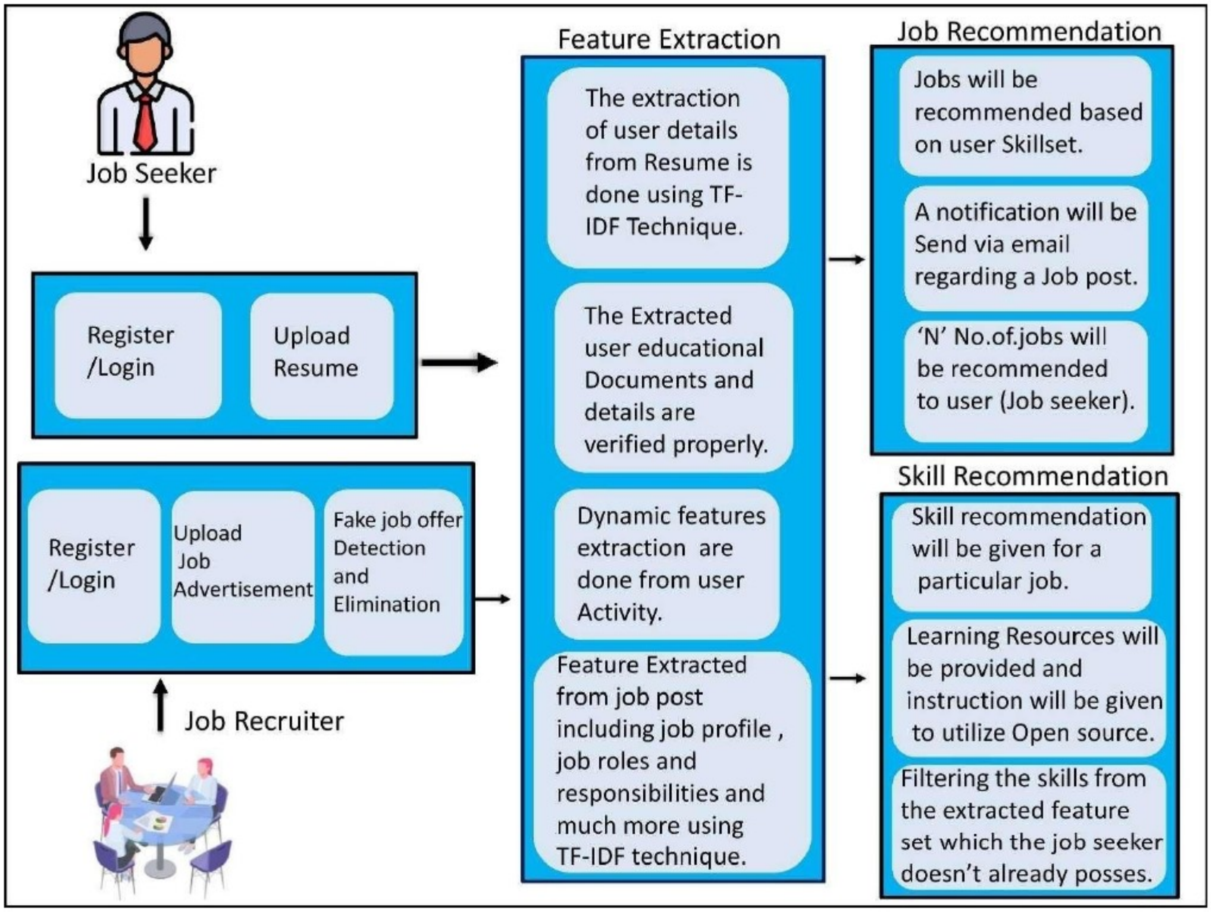

USER
FOR NEW USERS
INITIAL REGISTRATION
DATABASE
SIGN IN/LOGIN
SKILL/QUALIFICATION REGISTARTION
TO START THE JOB APPLICATION PROCESS
USE CHATBOT TO SOLVE QUERIES
JOB APPLICATION USING REG DETAILS
USE CHATBOT TO SOLVE QUERIES
LOGOUT

It shows how data enters and leaves the system, what changes the information, and where data is stored.

The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

5.2.SOLUTION & TECHNICAL ARCHITECTURE:

Solution architecture:



The extraction of user details from Resume is done using TF-IDF Technique.

The Extracted user educational Documents and details are verified properly.

Dynamic features extraction are done from user Activity.

Feature Extracted from job post including job profile , job roles and responsibilities and much more using TF-IDF technique.

Jobs will be recommended based on user Skillset.

A notification will be Send via email regarding a Job post.

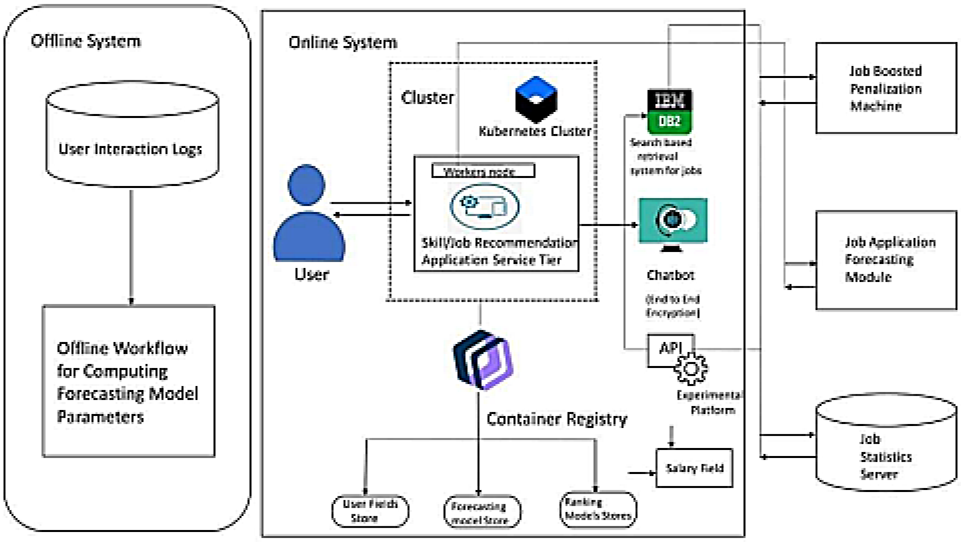
'N' No.of.jobs will be recommended to user (Job seeker).

Skill recommendation will be given for a particular job.

Learning Resources will be provided and instruction will be given to utilize Open source.

Filtering the skills from the extracted feature set which the job seeker doesn't already posses.

Technical architecture:



S. No	Component	Description	Technology
1	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc	HTML, CSS, JavaScript / Angular Js / React Js etc
2	Developing Interface	Developing application for the tas	Java / Python
3	Voice Assistance	Voice commands instead of typing	IBM Watson STT service
4	Chatbot Assistance	Conversational Interface	IBM Watson Assistant
5	Database	Data Type, Configurations etc	MySQL, NoSQL, etc
6	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc
7	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local File system
8	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc
9	Infrastructure (Server	Deployment on Local	Local, Cloud Foundry,

	/ Cloud)	System / Cloud Local Server Configuration: Cloud Server Configuration	Kubernetes, etc.

5.3.USER STORIES :

User Type	Functional Requirement (Epic	User Story / Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register 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 register ed 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 Facebo ok	I can register & access the dashboard with Facebo ok Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Facebo ok	I can receive confirmation email & click confirm	M**edium**	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sprint-1
	Dashboard	USN-6	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	Sprint-1
Customer (Web user	Identityaware	USN-7	Open, public access, Userauthe	Company public website. App	High	High

			ntic ated access, Employeerestricted access	running on the company intranet. App with access to customer private information		
Customer Care Executi ve	Communication	USN-8	customer care executive is a professional responsible for communicating the how's and why's regarding service expectations	For how to tackle customer queries	Medium	Sprint-1
Administrator	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Rem	Ease of use	Medium	Sprint-1

			ove Users in the directory.			

CHAPTER-6

PROJECT PLANNING &SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION:

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	**Likesh Kiumar BM,Gowtham K,Kishore K,Prasanth V**
Sprint-1		USN - 2	As a user, I will receive confirmation email	I can receive confirmation email & click	High	**Likesh Kumar B M ,Gowtham K**

			once I have register ed for the application	confirm		
Sprint-2		USN - 3	As a user, I can register for the application through Facebo ok	I can register & access the dashboard with Facebo ok Login	Low	**Likesh Kumar B M,Gowtham K,Kishore K**
Sprint-3		USN - 4	As a user, I can register for the application on throug Gmaiil	I can receive confirmation email & click confirm	Medium	**- M,Kishore K,Gowtham K ,Prasanth V**
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**	**Likesh Kumar B M, Prasanth V**
Sprint-2	Dashboard	USN - 6	Create a model set that contains those models, then assign it	Assign that group to the appropriate roles on the Roles page	High	L**ikesh Kumar B M,Gowtham K**

			to a role			

Sprint-4	IdentityAware	USN - 7	Open, public access, User authenticated access, Employe e restric ted access	Company public website. App running on the company intranet. App with access to customer private information	High	Gowtham K,Kishore K,Prasanth 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	Prasanth V ,Kishore K,Likesh Kumar B M

Sprint-3	Device management	USN -9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the director	Ease of use.	Medium	Likesh Kumar B M,Kishore K,Gowtham K,Prasanth V

6.2 Sprint Delivery Schedule:

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**

7.CODING & SOLUTIONING

STYLE.CSS:

```
body
{
   font-family: Century Gothic;
   margin: 0%;
}
ul
{
   list-style-type: none;
   margin: 0%;
   padding: 0;
   overflow: hidden;
}
li
{
   margin-left: 30px;
   display: inline;
   float: left;
}
li a
{
  display: block;
  color: hsl(0, 71%, 3%);
  text-align: center;
  padding: 14px 16px;
  text-decoration: none;
}
header
{
   width:100%;
   height: 100px;
   background-color:white(240, 240, 234);
   font-size: 20px;
```

```
}
li:hover
{
   background-color: rgb(97, 220, 159);
}

.two
{
   font-size: 35px;
   margin-top: 20px;
}
#sss1
{
   top: 50px;
   width: 100%;
   height: 80px;
   background-color: rgb(60, 88, 232);
   position: absolute;
}
.sec_nav
{
   margin-left: 900px;
}
.input_search
{
   margin-top: -73px;
   margin-left: 250px;
   width: 500px;
   height: 42px;
   border-radius: 5px;
   border: none;
   position: absolute;
   padding-left: 20px;
}
.button_search
{
   position: absolute;
```

```
    margin-top: -73px;
    height: 44px;
    margin-left: 790px;
    width: 80px;
    background-color: rgb(226, 14, 14);
    border-radius: 5px;
    font-family: Century Gothic;
}
.button_search:hover
{
    background-color: rgb(11, 10, 10);
    color: whitesmoke;
}
footer
{
    height: 30px;
    background-color: rgb(255, 63, 63);
    width: 100%;
}
.head {
    height: 50px;
    width: 100%;
    color: whitesmoke;
    background-color: black;
    font-size: 30px;
    font-weight: bold;
    padding-left: 20px;
    padding-top: 10px;
}



.head-git{
    height: 40px;
    width: 100%;
    justify-content: center;
    align-items: center;
    background-color: rgb(250, 244, 244);
    display: flex;
    flex-flow: row;
    margin-bottom: 5px;
  }
  .head-git h2 {
    font-size: 18px;
    font-family: 'Gill Sans', 'Gill Sans MT', Calibri, 'Trebuchet MS', sans-serif;
    color:black;
  }
  .head-git h2 a {
    text-decoration: none;
    color:blue;
    font-size: 23px;
  }
  .head-git h2 a:hover {
    color:rgb(211, 47, 41);
    background-color: beige;
  }

  .wrapper {
    height: 200px;
    width: 100%;
    margin-left:150px;
    display: flex;
    padding: 10px;
}
.wrapper .item {
    height: 200px;
    min-width: 20%;
    border-radius: 20px;
    margin: 10px;
    box-shadow: 10px 10px 10px rgba(0, 0, 0, 0.4);
    padding: 0.5px;
    border-bottom: 1px black;
}
.wrapper .item img {
    height: 100%;
    width: 100%;
}
.c-box {
    height: content;
    max-width: 50%;
    display: flex;
    padding: 10px;
    flex-flow: column;
    padding: 1px;
}
.c-box .content {
    height: content;
    min-width: 30%;
    display: flex;
    flex-flow: row wrap;
    margin-left: 130px;
    text-decoration: none;
}
.c-box .content p {
    color: black;
    font-size: 15px;
    font-weight: bold;
}
.c-box .content h2 {
    color: black;
    font-size: 20px;
    font-weight: bold;
}
.head-f{
    height: 40px;
    width: 100%;
    justify-content: center;
    border-top: 2px solid black;
    display: flex;
    flex-flow: column;
  }
```

```
.head-f h2 {
  font-size: 15px;
  color:black;
}

/*register and login css*/

/* Add padding to containers */
.container {
  padding: 16px;
  background-color: white;
}

/* Overwrite default styles of hr */
hr {
  border: 1px solid #f1f1f1;
  margin-bottom: 25px;
}

/* Set a style for the submit button */
.submitbtn {
  background-color: rgb(211, 47, 41);
  color: white;
  padding: 16px 20px;
  margin: 8px 0;
  border: none;
  cursor: pointer;
  opacity: 0.9;
  border-radius: 10px;
}

.submitbtn:hover {
  opacity: 1;
}

/* Add a blue text color to links */
a {
    color: dodgerblue;
  }
.form_label
{
  margin-left: 35%;
}
.form_input
{
  margin-left: 35%;
  width: 500px;
  height: 42px;
  padding-left: 20px;
  border-radius: 10px;
  border: none;
  background-color: whitesmoke;
}
```

HOME.HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SKYS.com</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
               <li><a href="{{url_for('home')}}">Home</a></li>
               <li><a href="{{url_for('register')}}">Register</a></li>
               <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
```

```
  </header>
  <section id="sss1">
    <h1 class="two" ><span style="color:rgb(109, 73, 214);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS</span>.com</h1>
    <div id="ss1">
      <input class="input_search" type="text" placeholder="Search Jobs" >
      <button class="button_search">Search</button>
    </div>
  </section>



</body>
</html>
```

REGISTER.HTML:

```
<html>
  <head>
    <title>Registration page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
      </ul>
    </nav>
  </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(109, 50, 237);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS</span>.com</h1></center>
    </section>
    <div class="container">
    <form action="/register" method="POST">
      <h2>Registration</h2>
      <label class="form_label"for="email"><b>Email ID</b></label><br><br>
      <input class="form_input"type="email" name="email"/><br><br>
      <label class="form_label"for="user"><b>Username</b></label><br><br>
      <input class="form_input"type="text" name= "username" /><br><br>
      <label class="form_label"for="psw"><b>Password</b></label><br><br>
      <input class="form_input"type="password" name="password"/><br><br>
      <label class="form_label"for="pho"><b> Enter Phone
number:</b></label><br><br>
      <input class="form_input"type="text" name="phonenumber"/><br><br>
      <label class="form_label"for="pho"><b> Enter four digit pin:</b></label><br><br>
      <input class="form_input"type="password" name="pin"/>
      </br></br></br>
     <center> {% if error %}
    <p><strong style="color:red">Error</strong>: {{error}}</p>
  {% endif %}
  {% with messages = get_flashed_messages() %}
     {% if messages %}
        {% for message in messages %}
          <p style="color:green">{{ message }}</p>
        {% endfor %}
     {% endif %}
   {% endwith %} </center>
      <center><input type="submit" class="submitbtn"value="submit" /></center>
      <center><p>Already have a account <a href="{{url_for('login')}}">Sign
in</a></p></center>
    </form>
    </div>
  </body>
</html>
```

LOGIN.HTML:

```
<html>
  <head>
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
      </ul>
    </nav>
    </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(109, 60, 225);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS</span>.com</h1></center>
    </section>
    <br><br>
    <div class="container"> <br /><br />
    <form action="/login" method="POST">
     <h1>Login</h1> <br /><br />
      <label class="form_label"for="email"><b>Username</b></label><br><br>
      <input class="form_input"type="text" name= "username" /><br><br>
      <label class="form_label"for="psw"><b>Password</b></label><br><br>
      <input class="form_input"type="password" name="password"/>
      </br></br></br>
      <center><input type="submit" class="submitbtn"value="submit" /></center>
    </form>
    </div>
    <center><p>Forgot your password? <a href="{{url_for('forget')}}">Try Another
Way</a></p></center>
    <center><p>Don't have a account <a href="{{url_for('register')}}">Create new
account</a></p></center>
  </body>
</html>
```

FORGET.HTML:

```
<html>
  <head>
```

```
    <title>Login page</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
  </head>
  <body>
    <header>
    <nav class="navbar">
      <ul>
        <li><a href="{{url_for('welcome_page')}}">Back to page</a></li>
      </ul>
    </nav>
    </header>
    <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(109, 30, 192);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS</span>.com</h1></center>
    </section>
    <br><br>
  {% if error %}
    <p><strong style="color:red">Error</strong>: {{error}}</p>
  {% endif %}
  {% with messages = get_flashed_messages() %}
     {% if messages %}
        {% for message in messages %}
          <p style="color:green">{{ message }}</p>
        {% endfor %}
     {% endif %}
   {% endwith %}
    <div class="container"> <br /><br />
    <form action="/forget" method="POST">
     <h1>Try to login with your 4 digit pin</h1> <br /><br />
      <label class="form_label"for="email"><b>Username</b></label><br><br>
      <input class="form_input"type="text" name= "username" /><br><br>
      <label class="form_label"for="psw"><b>Pin</b></label><br><br>
      <input class="form_input"type="password" name="pin"/>
      </br></br></br>
      <center><input type="submit" class="submitbtn"value="submit" /></center>
    </form>
```

```
        </div>
        <center><p>Back to <a href="{{url_for('login')}}">login</a></p></center>
    </body>
</html>
```

WELCOME.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SKYS.com</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{url_for('register')}}">Register</a></li>
                <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(109, 100, 237);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS</span>.com</h1></center>
    </section>
</body>
</html>
```

ABOUT.HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SKYS.com</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                 <li><a href="{{url_for('home')}}">Home</a></li>
                <li><a href="{{url_for('register')}}">Register</a></li>
                <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS.com</span></h1></center>
    </section>
</body>
</html>
```

CONTACT.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SKYS.com</title>
    <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
```

```
</head>
<body>
    <header>
        <nav class="navbar">
            <ul>
                <li><a href="{{url_for('home')}}">Home</a></li>
               <li><a href="{{url_for('register')}}">Register</a></li>
               <li><a href="{{url_for('login')}}">Login</a></li>
            </ul>
        </nav>
    </header>
    <section id="sss1">
        <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS.com</span></h1></center>
    </section>
    <br><br>
    <center><h1 class="h1lt">Contact us</h1></center>

    <form>
 First name:<br>
 <input type="text" name="firstname" value="">
 <br>
 Last name:<br>
 <input type="text" name="lastname" value="">
 <br>
 Email:<br>
 <input type="text" name="email" value="">
 <br>
 <br>
  <textarea name="message" rows="10" cols="30">
</textarea>
 <br><br>
 <input type="submit" value="Submit">
</form>

</body>
```

```
</html>
```

SKILLS.HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>SKYS.com</title>
   <link rel="stylesheet" href="{{url_for('static', filename='style.css')}}">
</head>
<body>
   <header>
      <nav class="navbar">
         <ul>
             <li><a href="{{url_for('home')}}">Home</a></li>
            <li><a href="{{url_for('register')}}">Register</a></li>
            <li><a href="{{url_for('login')}}">Login</a></li>
         </ul>
      </nav>
   </header>
   <section id="sss1">
      <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:
whitesmoke;margin-left: 30px;padding-left: 20px;padding-right:
20px;">SKYS.com</span></h1></center>
   </section>
   <br><br>
   <section id="courses">

                                             <center><h1>Our Courses</h1></center>

                                             <div class="course">
         <img src="{{url_for('static',filename='img/course-01.jpg')}}"alt="skill 1">
```

```
                                        <center><h2><a
href="https://www.w3schools.com/html/">HTML5 For Beginners</a></h2></center>

                                        <center><h6>This course was designed for
students starting out in Front End Web Development wanting to learn HTML5 to get
started........</h6></center>



                                   </div>

                                   <div class="course2">
     <img src="{{url_for('static',filename='img/course-02.jpg')}}"alt="skill 2">


                                        <center><h2><a
href="https://www.w3schools.com/css/">CSS3 For Beginners</a></h2></center>

                                        <center><h6>This course was designed for
students starting out in Front End Web Development wanting to learn CSS3 to get
started........</h6></center>



                                   </div>

                                   <div class="course3">

                                        <img
src="{{url_for('static',filename='img/course-03.jpg')}}"alt="skill 3">

                                        <center><h2><a
href="https://www.w3schools.com/js/">JavaScript For Beginners</a></h2></center>

                                        <center><h6>This course was designed for
students starting out in Front End Web Development wanting to learn JavaScript to get
started........</h6></center>
```

```
                                          </div>

                                        </section>

</body>
</html>
```

APP.PY

```
from flask import Flask, render_template,request,redirect,url_for,session,flash
import ibm_db
import os
from sendgrid import SendGridAPIClient
from sendgrid.helpers.mail import Mail
import requests

app=Flask(__name__)
app.secret_key='a'
try:
  conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=98538591-7217-4024-b027-
8baa776ffad1.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=30875;SEC
URITY=SSL;SSLServerCertificat=DigiCertGlobalRootCA.crt;UID=chr84960;PWD=ky5p7N
0JrFc3Ibvx",",")
except:
  print("Unable to connect: ",ibm_db.conn_error())

@app.route("/")
def dash():
  return render_template('welcome.html',msg=" ")

@app.route("/register",methods=['GET','POST'])
def register():
  error = None
  if request.method=='POST':
        username=request.form['username']
        email=request.form['email']
        phone_number=request.form['phonenumber']
        password=request.form['password']
        pin=request.form['pin']
        sql="SELECT * FROM user WHERE phone_number=?"
        prep_stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(prep_stmt,1,phone_number)
        ibm_db.execute(prep_stmt)
        account=ibm_db.fetch_assoc(prep_stmt)
        print(account)
          #message = 
Mail(from_email='btechmano@gmail.com',to_emails=session['email'],subject="Devnews 
- Registration",html_content='<b>Devnews welcomes you</b><br/><p>Your account has 
been registered successfully</p>')
         #try:
            #sg=SendGridAPIClient()
            # Secret key can't be submitted otherwise my
            # sendgrid accound reporting that i am exposing
            # my secret key as public and my account will terminated soon
            #response=sg.send(message)
            #print(response.status_code)
            #print(response.body)
            #print(response.headers)
         #except Exception as e:
            #print(e)
        if account:
           error="Account already exists! Log in to continue !"
        else:
           insert_sql="INSERT INTO user values(?,?,?,?,?)"
           prep_stmt=ibm_db.prepare(conn,insert_sql)
           ibm_db.bind_param(prep_stmt,1,email)
           ibm_db.bind_param(prep_stmt,2,username)
           ibm_db.bind_param(prep_stmt,3,phone_number)
           ibm_db.bind_param(prep_stmt,4,password)
           ibm_db.bind_param(prep_stmt,5,pin)
           ibm_db.execute(prep_stmt)
```

```
                flash(" Registration successfull. Log in to continue !")
    else:
        pass
    return render_template('register.html',error=error)

@app.route('/login',methods=['GET','POST'])
def login():
    error = None
    if request.method=='POST':
        username=request.form['username']
        password=request.form['password']
        sql="SELECT * FROM user WHERE username=? AND password=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session["username"]=account["USERNAME"]
            flash("Logged in successfully!")
            return redirect(url_for("home"))
        else:
            error="Incorrect username / password"
            return render_template('login.html',error=error)
    return render_template('login.html',error=error)

@app.route('/forget',methods=['GET','POST'])
def forget():
    error = None
    if request.method=='POST':
        username=request.form['username']
        pin=request.form['pin']
        sql="SELECT * FROM user WHERE username=? AND pin=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,username)
        ibm_db.bind_param(stmt,2,pin)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        print(account)
        if account:
            session['Loggedin']=True
            session['id']=account['USERNAME']
            session["username"]=account["USERNAME"]
            flash("Logged in successfully!")
            return redirect(url_for("home"))
        else:
            error="Incorrect username / pin"
            return render_template('login.html',error=error)
    return render_template('forget.html',error=error)
@app.route('/welcome')
def welcome_page():
    return render_template("welcome.html",msg=" ")
@app.route('/home')
def home():
    return render_template("home.html",msg=" ")
@app.route('/skills')
def skills():
    return render_template("skills.html",msg=" ")
@app.route('/about')
def about():
    return render_template("about.html",msg=" ")
@app.route('/contact')
def contact():
    return render_template("contact.html",msg=" ")
if __name__=='__main__':
    app.run(debug=True)
```

8.TINGTES

8.1 TEST CASES:

TestcaseID	FeatureType	Component	TestScenario
LoginPage_TC_O O1	Functional	HomePage	Verifyuser is able to see theLogin/Signup popup when userclickedonMy accountbutton
LoginPage_TC_O O2	UI	HomePage	Verify the UI elements inLogin/Signuppop up
LoginPage_TC_O O3	Functional	HomePage	Verify user is able to log intoapplicationwit hV alidcredenti als
LoginPage_TC_O O4	Functional	Loginpage	Verify user is able to log intoapplicationwithI nValidcredenti also
LoginPage_TC_O O5	Functional	Loginpage	Verify user is able to log intoapplicationwithI nValidcredenti also

ExpectedResult	ActualResult	Status	Commnets
Login/Signuppopup shoulddisplay	Working asexpected	pass	
Application should show below	Working asexpected	pass	

UIelements: a.email text boxb.passwordtext box c.Login button with orange colourd.New customer? Create account linke.Last passw			
User should navigate to user accounthomepage	Working asexpected	pass	
Application should show 'Incorrectemail or password ' validationmessage.	Working asexpected	pass	
Application should show 'Incorrectemail or password ' validationmessage	Working asexpected	pass	
Application should show 'Incorrectemail or password ' validationmessage	Working asexpected	pass	

<u>8.2.USER ACCEPTANCE TESTING</u>:

<u>PurposeofDocument:</u>

Thepurposeofthisdocumentistobrieflyexplainthetestcoverageandopenissuesofthe[ProductName]projectatthetimeofthereleasetoUserAcceptance Testing (UAT)

<u>DefectAnalysis:</u>

Thisreportshowsthenumberofresolvedor closedbugsateachseverity level,andhowthey wereresolved.

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
ByDesign	10	4	2	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
NotReproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won'tFix	0	5	2	1	8
Totals	24	14	13	16	77

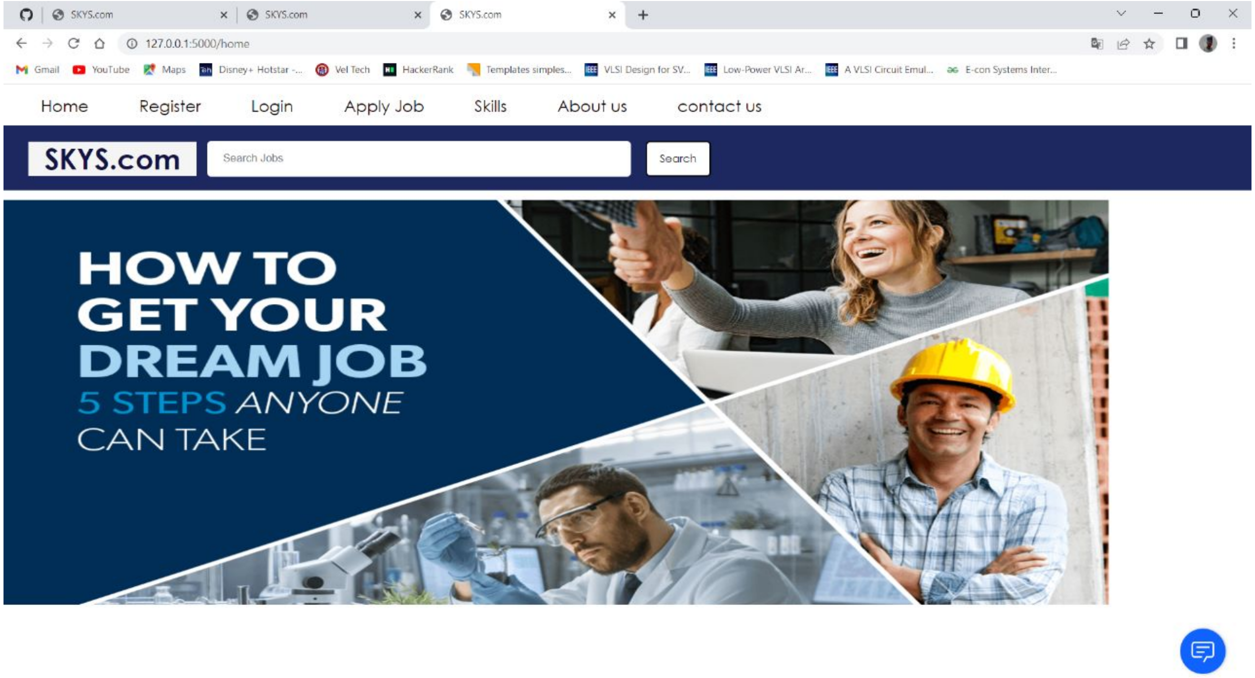
TestCaseAnalysis:

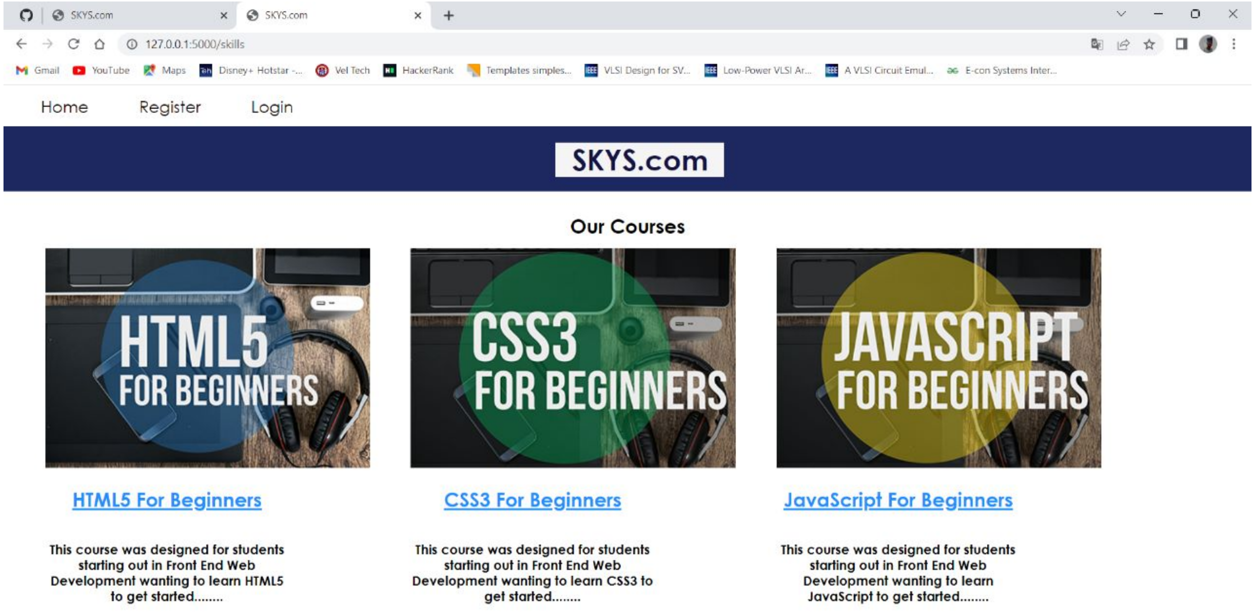
Thisreportshowsthenumberoftestcasesthathavepassed,failed,anduntested

Section	TotalCases	NotTested	Fail	Pass
PrintEngine	7	0	0	7
ClientApplication	51	0	0	51
Security	2	0	0	2
OutsourceShipping	3	0	0	3
ExceptionReporting	9	0	0	9
FinalReportOutput	4	0	0	4
VersionControl	2	0	0	2

9.RESULTS

9.1.PERFORMANCE METRICS:





Home Register Login

SKYS.com

Our Courses

HTML5 For Beginners

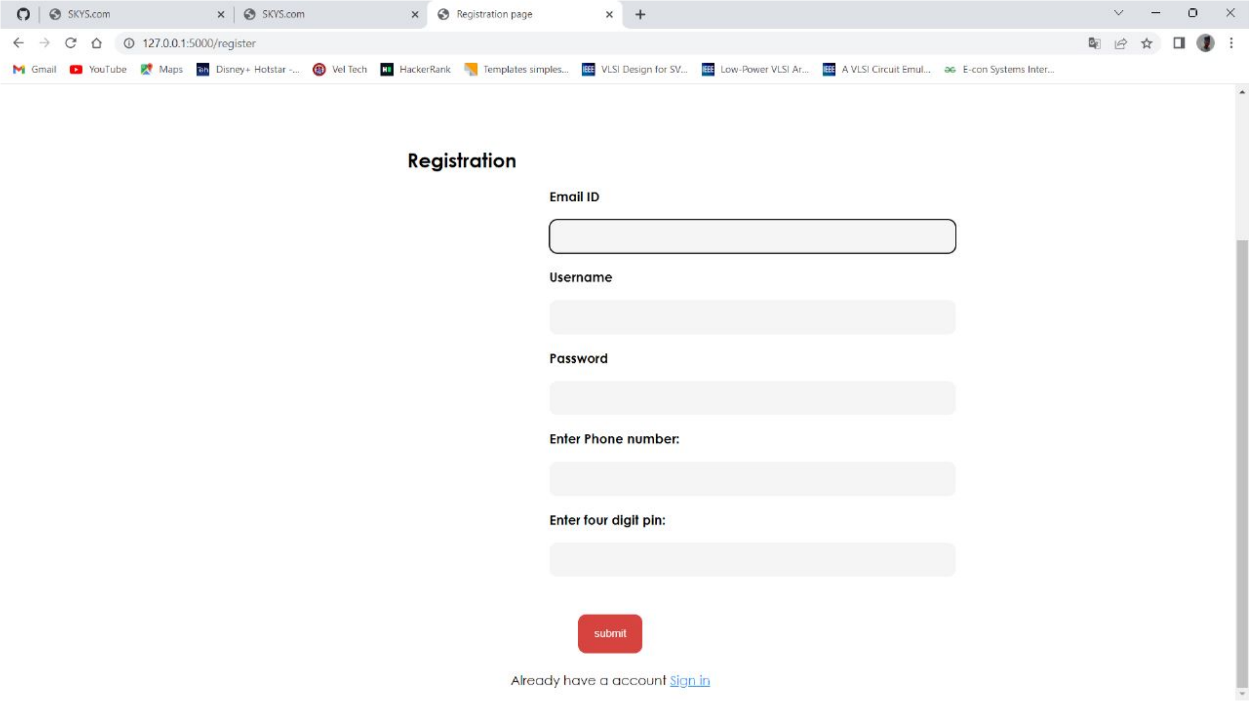
This course was designed for students starting out in Front End Web Development wanting to learn HTML5 to get started........

CSS3 For Beginners

This course was designed for students starting out in Front End Web Development wanting to learn CSS3 to get started........

JavaScript For Beginners

This course was designed for students starting out in Front End Web Development wanting to learn JavaScript to get started........



Registration

Email ID

Username

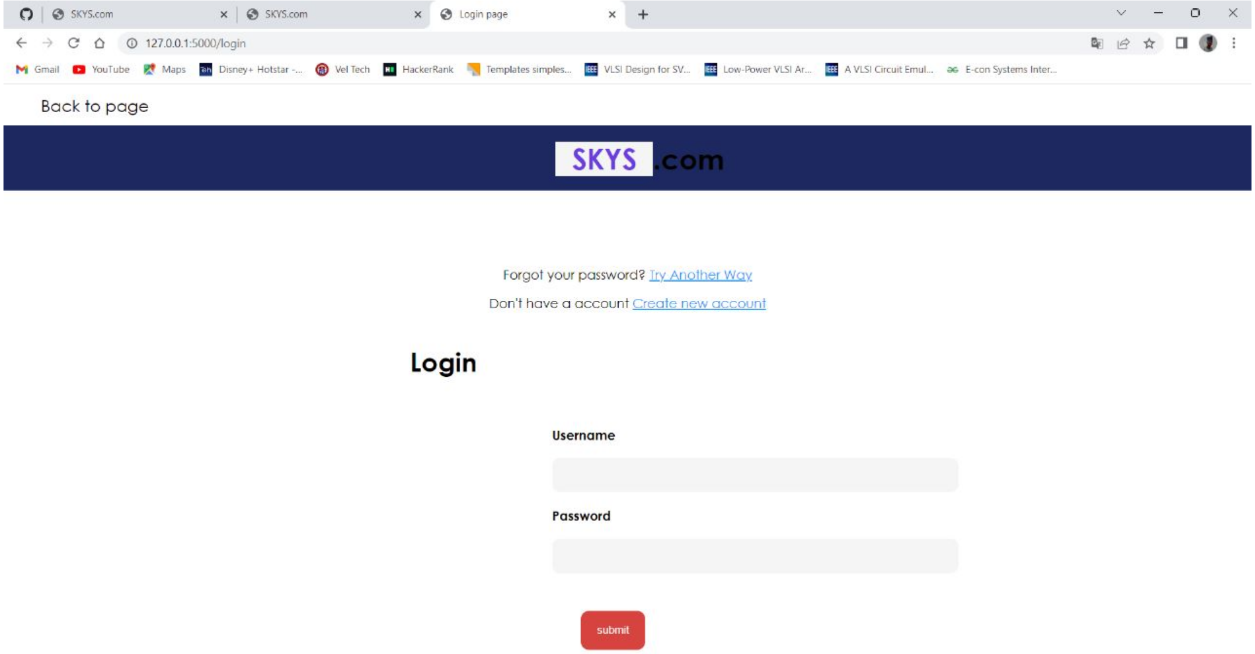
Password

Enter Phone number:

Enter four digit pin:

submit

Already have a account Sign in



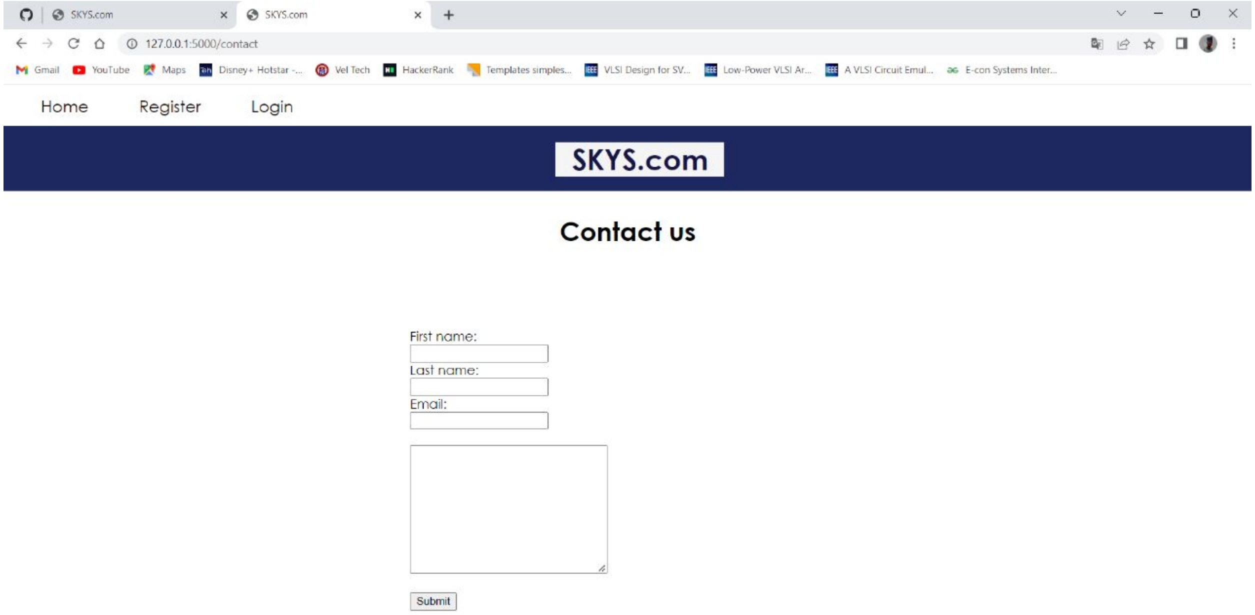
Forgot your password? Try Another Way

Don't have a account Create new account

Login

Username:

Password:



Contact us

First name:

Last name:

Email:

10.CONCLUSION

Job Recommendation System has a major role to play among recommending systems. With the presence of new algorithms and techniques, the system needs to evolve along with it. The main objective of this project is to recommend a suitable job for the candidates. This project has two pre-processing methods, one text mining method and one similarity function. The pre-processing methods are stop words and porter stemmer. The text mining method is tf-idf. The similarity function is a cosine similarity function. Pre-processing methods are used with resumes and with jobs description, to make the system more efficient by avoiding some garbage words. Tf-idif is used in processed resumes and processed jobs descriptions to convert it from text to matrix to compare. Cosine Similarity will measure the similarity between the resume and each job description.

Finally, it will display the scores for the jobs in a sorted way. There is also a pie chart which is used to visualize the percentage of the scores which is got by the candidate for the jobs. Then use a list compare 58 method to compare the resume and job skills to recommend the skills to be improved by the candidate

11.REFERENCE

[1] R. J. Mooney and L. Roy, "Content-Based Book Recommending Using Learning for Text Categorization," in In Proceedings of DL ’00: Proceedings of the Fifth ACM Conference on Digital Libraries, New York, NY, pp. 13-20, 2000.

[2] Li-Ping Jing, Hou-Kuan Huang, Hong-Bo Shi, “Improved feature selection approach TFIDF in text mining”, International Conference on Machine Learning and Cybernetics, pp. 944-946, 2002, doi:10.1109/icmlc.2002.1174522.

[3] Shouning Qu ,Sujuan Wang,Yan Zou, ” Improvement of Text Feature Selection Method Based on TFIDF”, International Seminar on Future Information Technology and Management Engineering, pp. 79-81, 2008, doi:10.1109/fitme.2008.25.

[4] I. A. Braga, "Evaluation of stopwords removal on the statistical approach for automatic term extraction," Seventh Brazilian Symposium in Information and Human Language Technology, pp. 142-149, 2009.

[5] Nikolaos D. Almalis, Prof. George A. Tsihrintzis, Nikolaos Karagiannis, Aggeliki D. based job recommendation algorithm for job seeking and recruiting”, 6th International Conference on

Information, Intelligence, Systems and Applications (IISA), pp. 1-7, 2015, doi:10.1109/iisa.2015.7388018.

[6] Mohammad Alodadi and Vandana P. Janeja, “ Similarity in Patient Support Forums Using TF-IDF and Cosine Similarity Metrics”, International Conference on Healthcare Informatics, pp. 521-522, 2015, doi:10.1109/ichi.2015.99.

[7] L. Zahrotun, “Comparison jaccard similarity, cosine similarity and combined both of the data clustering with shared nearest neighbor method,” Computer Engineering and Applications Journal. vol. 5. Pp. 11- 18, 2016, doi:10.18495/comengapp.v5i1.160, 2016.

[8] Peng Yi, Cheng Yang ,Chen Li, Yingya Zhang, “A Job Recommendation Method Optimized by Position Descriptions and Resume Information”, IEEE Advanced Information Management, Communicates, Electronic and Automation 59 Control Conference (IMCEC), pp. 762 -764, March 2017, doi:10.1109/rteict.2017.8256590.

[9] Minh-Luan Tran, Anh-Tuyen Nguyen, Quoc-Dung Nguyen, Tin Huynh, “A comparison study for job recommendation”, International Conference on Information and Communications (ICIC), pp. 199-204, 2017, doi:10.1109/infoc.2017.8001667.

[10] Gokul P.P, Akhil BK, Shiva Kumar K.M, “Sentence similarity detection in Malayalam language using cosine similarity”, 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 221-225, 2017, doi:10.1109/rteict.2017.8256590.

[11] Leah G. Rodriguez, Enrico P. Chavez, “Feature Selection for Job Matching Application using Profile Matching Model”, IEEE 4th International Conference on Computer and Communication Systems (ICCCS), pp. 2-4 , FebStrati.

[12] Nunik Destria Arianti, Mohamad Irfan, Undang Syaripudin, Dina Mariana, Neny Rosmawarni, Dian Sa’adillah Maylawati, “Porter Stemmer and Cosine Similarity for Automated Essay Assessment”, 5th International Conference on Computing Engineering and Design (ICCED)”, pp. 1-5, 2019, doi:10.1109/icced46541.2019.91610.

[13] Garima Koushik, Dr. Prof. K. Rajeswari, Mr. Suresh Kannan Muthusamy, “Automated Hate Speech Detection on Twitter. 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA)”, pp. 421- 425, 2019, doi:10.1109/iccubea47591.2019.912.

[14] Ravali Boorugu, Dr. G. Ramesh, “A Survey on NLP based Text Summarization for Summarizing Product Reviews”, Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 352- 356, 2020, doi: 10.1109/ICIRCA48905.2020.9183355

.

[15] Tanya V. Yadalam,Vaishnavi, M.Gowda, Vanditha Shiva Kumar, Disha Girish, Namratha M, “Career Recommendation System Using Content Based Filtering”, International Conference on Communication and Electronics Systems (ICCES), pp. 2-5, June 2020, doi: 10.1109/ICCES48766.2020.9137992

[16] Jeevamol Joy and Renumol V G, “Comparison of Generic Similarity Measures in E-learning Content Recommender System in Cold-Start Condition”, IEEE Bombay Section Signature Conference (IBSSC)”, pp. 60 175- 179, 2020, doi:10.1109/ibssc51096.2020.9332162.

[17] M. Alamelu, D.Sathish Kumar, R. Sanjana, J.Subha Sree, A.Sangeerani Devi, D. Kavitha, “Resume Validation and Filtration using Natural Language Processing”, 10th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), pp. 412-430, 2021, doi:10.1109/IEMECON53809.2021.9689075.

[18] Swaranjali Jugran, Ashish Kumar, Bhupendra Singh Tyagi, Mr. Vivek Anand,”Extractive Automatic Text Summarization using SpaCy in Python & NLP”, International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), pp. 582-585, 2021, doi:10.1109/icacite51222.2021.9404712.

[19] S. Prathyusha, S. Jadhav, K. Kommu, M.S. Velpuru, “Text summarization using NLTK with GUI interface”, 4th Smart Cities Symposium (SCS 2021), pp. 435-442, 2021, doi: 10.1049/icp.2022.0369.

[20] Meenakshi A. Thalor, “A Descriptive Answer Evaluation System Using Cosine Similarity Technique”, International Conference.

GITHUB LINK : https://github.com/IBM-EPBL/IBM-Project-21369-1659778736

PROJECT DEMO LINK : https://www.youtube.com/embed/ANU2OWyNrc8