Skill / Job Recommender Application

Team ID- PNT2022TMID52983

Batch-B8-2A4E

PROJECT REPORT

1. INTRODUCTION

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

1.1 PROJECT OVERVIEW

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

1.2 PURPOSE

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

2. LITERATURE SURVEY

Abstract

In this project, the job recommendation systems aim to minimize information overload by helping user’s in searching desired job information. Faced with this scenario, we investigate the use of cloud factors able to have a positive influence on generating recommendations. Thus, we present a new, simple model based on cloud features which is associated with the content-based technique of job recommendation. The practical applicability of data storage environments in the cloud provides the best use of cloud resources and meets user’s preferences.

In this project, We

- Made publicly accessible a new dataset formed by a set of job seekers profiles and a set of job vacancies amassed from distinctive job search engine sites
- Put ahead the proposal of a framework for job suggestion based totally on expert competencies of job seekers
- Carried out an assessment to quantify empirically the advice competencies of two state of-the-art methods, thinking about special configurations, inside the proposed framework.

Related Projects

1.TOPIC: Job Recommendation System Using Profile Matching And Web-Crawling

A Deepali V Musale , Mamta K Nagpure , Kaumudini S Patil , Rukhsar F Sayyed Students, Computer Science & Engineering, K K Wagh College of Engineering, Nashik, India

SUMMARY

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.

2.TOPIC : JOB SEARCH PORTAL

SOWMYA MATHUKUMALLI, SASTRA University, India, 2014

The web application "Job Search Portal" provides an easy and convenient search application for the job seekers to find their desired jobs and for the recruiters to find the right candidate. Job seekers from any background can search for the current job openings. Job seekers can register with the application and update their details and skill set. They can search for available jobs and apply to their desired positions. Android, being open source has already made its mark in the mobile application development. To make things handy, the user functionalities are developed as an Android application. Employer can register with the application and posts their current openings. They can view the Job applicants and can screen them according to the best fit. Users can provide a review about an organization and share their interview experience, which can be viewed by the Employers.

3.TOPIC : Enhanced Job Recommendation System

Shivraj Hulbatte, Amit Wabale , Suraj Patil , Nikhilkumar Sathe Student, Department of
Computer Engineering, Sinhgad Academy of Engineering, Pune, India

We address the problem of recommending suitable jobs to people who are seeking a new job. We formulate this recommendation problem as a supervised machine learning problem. Our technique exploits all past job transitions as well as the data associated with employees and institutions to predict an employee's next job transition. Dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. To reduce this laborious work, we design and implement a recommendation system for online job hunting. In this paper, we contrast user-based and item-based collaborative filtering algorithm to choose a better performed one. We also take background information including students' resumes and details of recruiting information into consideration, bring weights of co-apply users (the users who had applied the candidate jobs) and weights of student used liked jobs into their recommendation algorithm. At last, the model we proposed is verified through experiments study which is using actual data. The recommended results can achieve higher score of precision and recall, and they are more relevant with users' preferences.

4.TOPIC : APPLYING DATA MINING FOR JOB RECOMMENDATIONS BY EXPLORING JOB PREFERENCES

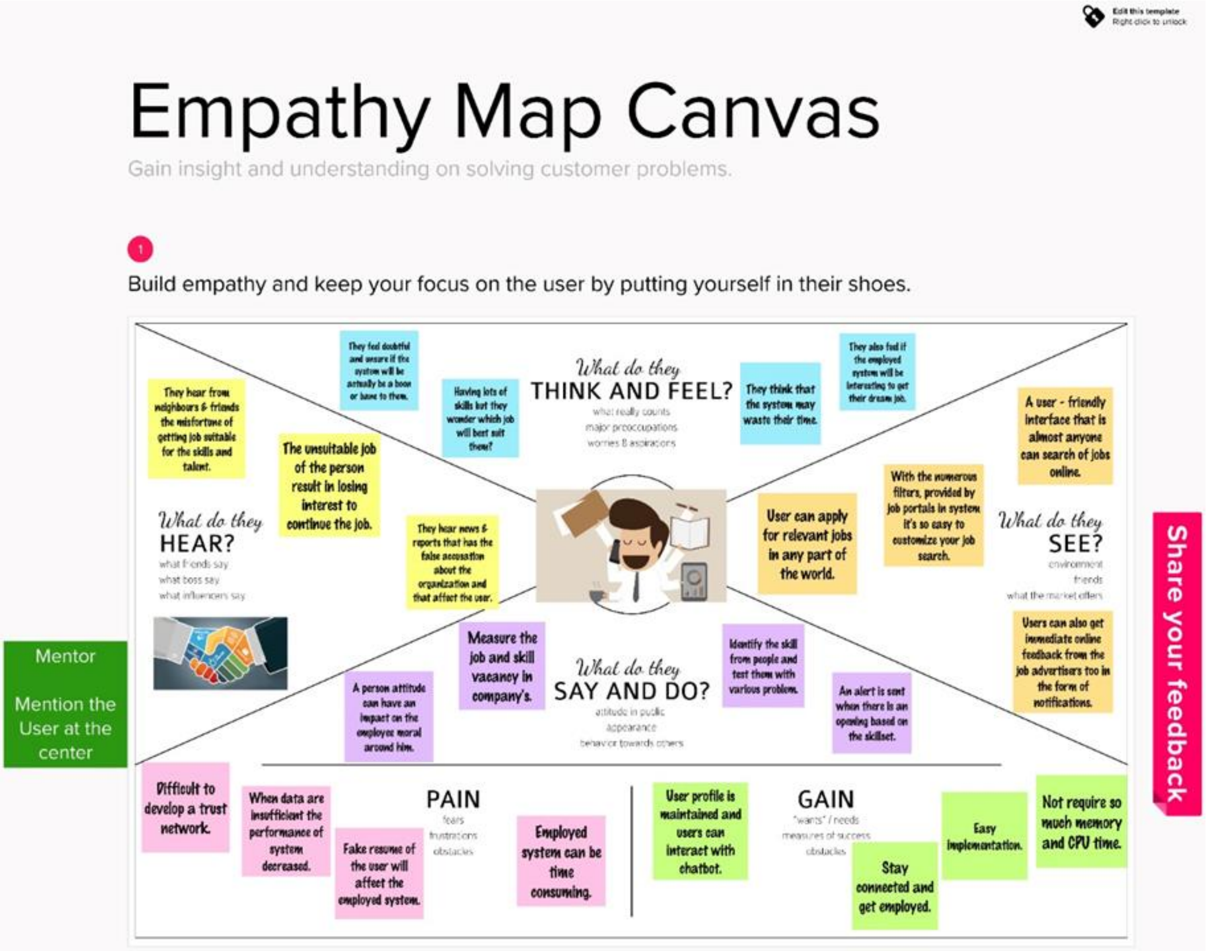
The recommender systems are quite popular as they help to find the customer what they want within a very less time. The recommendations are the guesses made by the system about an item that a customer will most likely prefer. These help to increase the site‟s popularity as well as sales (in case of business sites). Although there are generalized recommender systems, but personalized recommender systems are more focused upon. Personalized recommender systems are expected to change the content or items according to the user‟s profile and preferences. Analogous to the personalized recommender systems, generalized recommender systems provide same content to all the users. There are various types of recommender system strategies: Content Based, Collaborative Based, Demographic, Knowledge Based and Hybrid Recommender.

3. IDEATION AND PROPOSED SOLUTION:

3.1 Empathy Map

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

- Create a shared understanding of user needs, and
- Aid in decision making



Edit this template
Right click to unlock

Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

They feel doubtful and unsure if the system will be actually be a boon or bane to them.

Having lots of skills but they wonder which job will best suit them?

They think that the system may waste their time.

They also feel if the employed system will be interesting to get their dream job.

What do they HEAR?
what friends say
what boss say
what influencers say

They hear from neighbours & friends the misfortune of getting job suitable for the skills and talent.

The unsuitable job of the person result in losing interest to continue the job.

They hear news & reports that has the false accusation about the organization and that affect the user.

What do they SEE?
environment
friends
what the market offers

A user - friendly interface that is almost anyone can search of jobs online.

With the numerous filters, provided by job portals in system it's so easy to customize your job search.

User can apply for relevant jobs in any part of the world.

Users can also get immediate online feedback from the job advertisers too in the form of notifications.

What do they SAY AND DO?
attitude in public
appearance
behavior towards others

A person attitude can have an impact on the employee moral around him.

Measure the job and skill vacancy in company's.

Identify the skill from people and test them with various problem.

An alert is sent when there is an opening based on the skillset.

PAIN
fears
frustrations
obstacles

Difficult to develop a trust network.

When data are insufficient the performance of system decreased.

Fake resume of the user will affect the employed system.

Employed system can be time consuming.

GAIN
"wants" / needs
measures of success
obstacles

User profile is maintained and users can interact with chatbot.

Stay connected and get employed.

Easy implementation.

Not require so much memory and CPU time.

Mentor

Mention the User at the center

Share your feedback

3.2 IDEATION AND BRAINSTROMING:

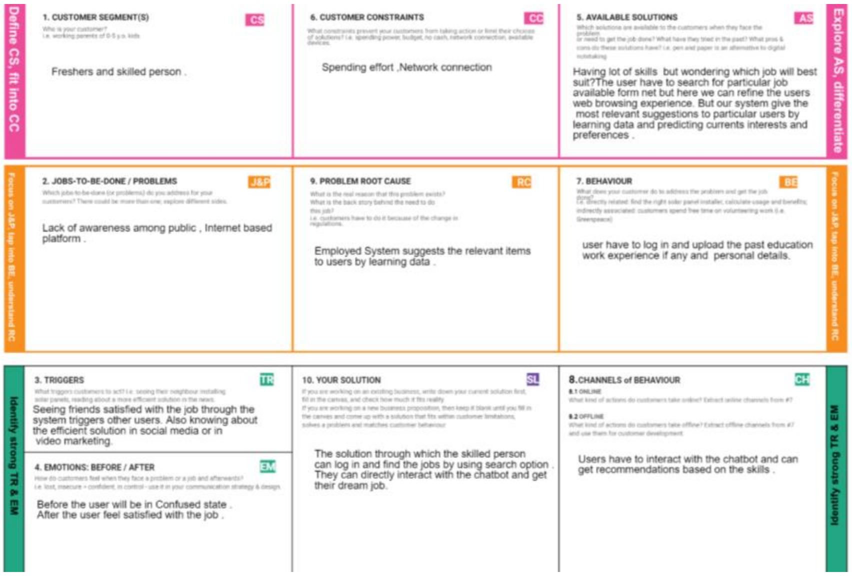
Brainstorm & Idea Prioritization Template:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of- the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

3.2 PROPOSED SOLUTION

Having lots of skills but wondering which job will best suit you? Don't need to worry! We have come up with a skill recommender solution through which the fresher or the skilled person can log in and find the jobs by using the search option or they can directly interact with the chatbot and get their dream job. To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. Wecan use a job search API to get the current job openings in the market which will fetch the data directly from the webpage .

3.3 PROBLEM SOLUTION FIT



Define CS, fit into CC
Explore AS, differentiate
Focus on J&P, tap into BE, understand RC
Identify strong TR & EM

1. CUSTOMER SEGMENT(S) CS

Who is your customer? i.e. working parents of 0-5 y.o. kids

Freshers and skilled person .

6. CUSTOMER CONSTRAINTS 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.

Spending effort ,Network connection

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Having lot of skills but wondering which job will best suit?The user have to search for particular job available form net but here we can refine the users web browsing experience. But our system give the most relevant suggestions to particular users by learning data and predicting currents interests and preferences .

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.

Lack of awareness among public , Internet based platform .

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.

Employed System suggests the relevant items to users by learning data .

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)

user have to log in and upload the past education work experience if any and personal details.

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.

Seeing friends satisfied with the job through the system triggers other users. Also knowing about the efficient solution in social media or in video marketing.

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.

Before the user will be in Confused state .
After the user feel satisfied with the job .

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.

The solution through which the skilled person can log in and find the jobs by using search option . They can directly interact with the chatbot and get their dream job.

8.CHANNELS of BEHAVIOUR CH

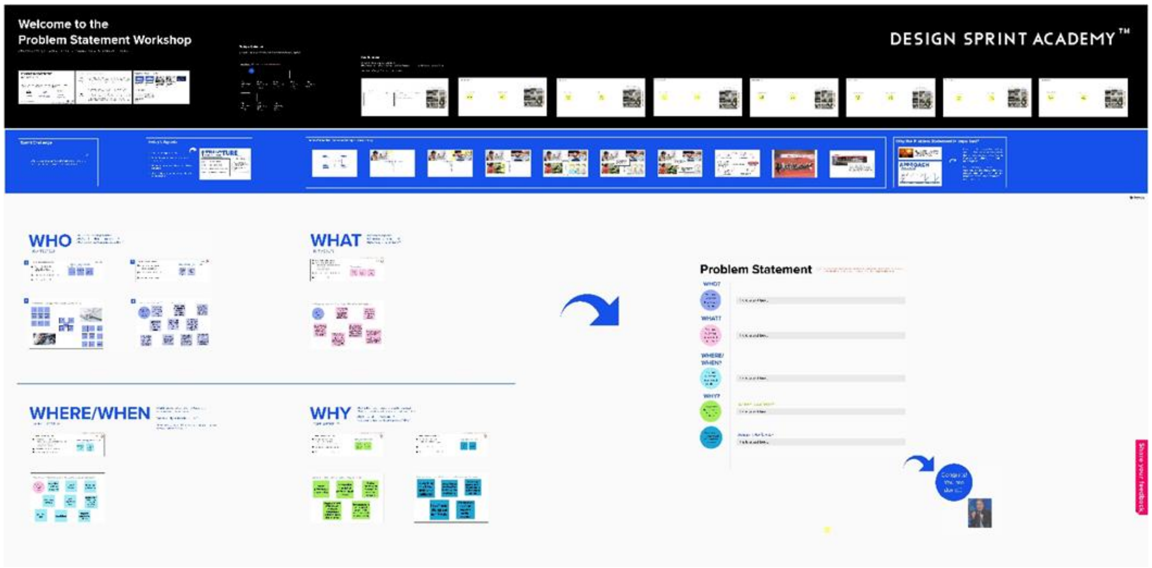
8.1 ONLINE

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

8.2 OFFLINE

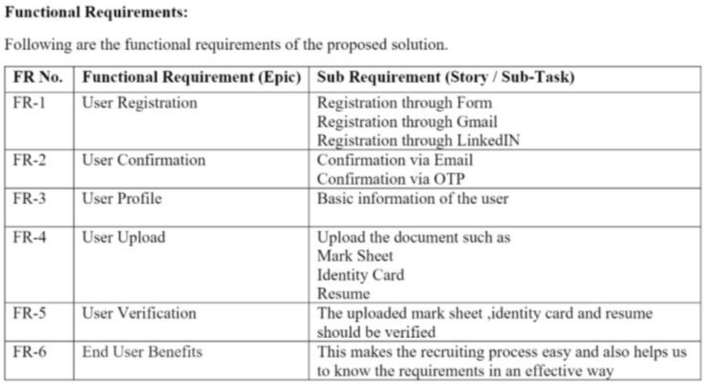
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Users have to interact with the chatbot and can get recommendations based on the skills .



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:

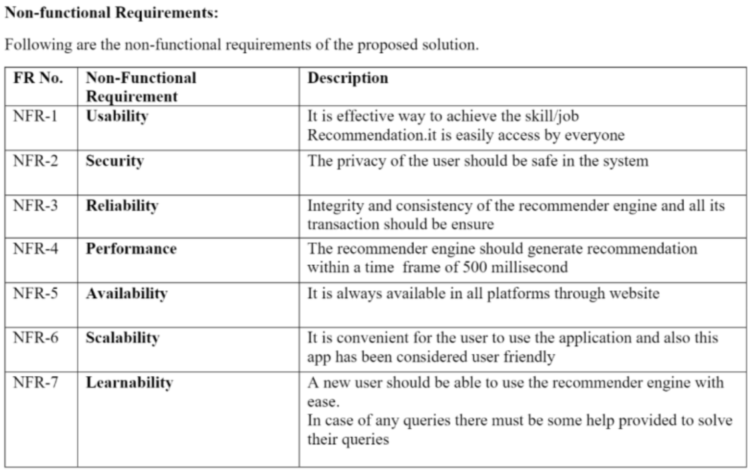


Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Profile	Basic information of the user
FR-4	User Upload	Upload the document such as Mark Sheet Identity Card Resume
FR-5	User Verification	The uploaded mark sheet ,identity card and resume should be verified
FR-6	End User Benefits	This makes the recruiting process easy and also helps us to know the requirements in an effective way

4.2 NON–FUNCTIONAL REQUIREMENTS:



Non-functional Requirements:

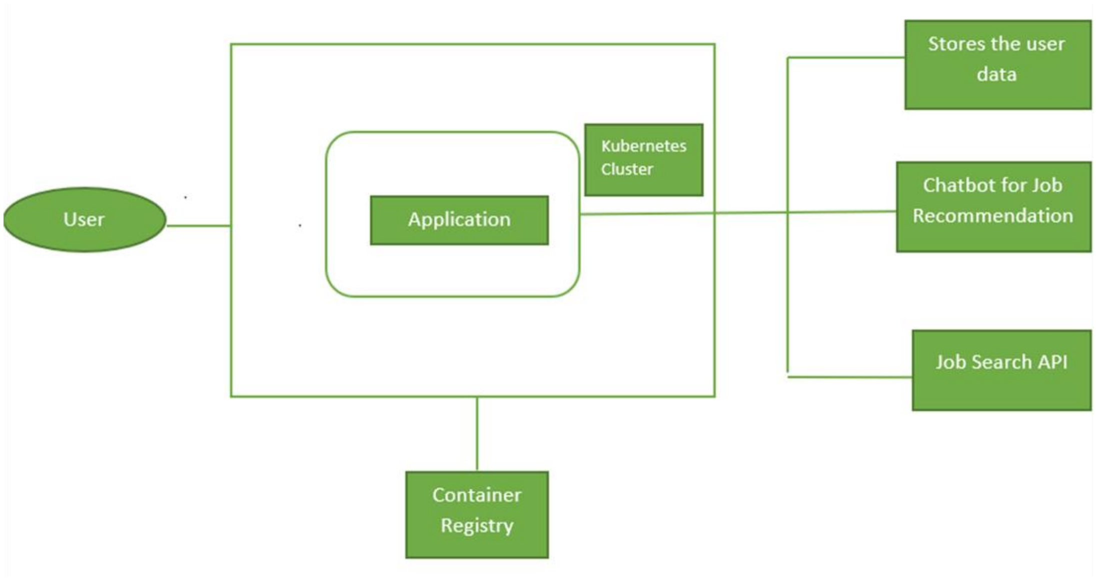
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It is effective way to achieve the skill/job Recommendation.it is easily access by everyone
NFR-2	**Security**	The privacy of the user should be safe in the system
NFR-3	**Reliability**	Integrity and consistency of the recommender engine and all its transaction should be ensure
NFR-4	**Performance**	The recommender engine should generate recommendation within a time frame of 500 millisecond
NFR-5	**Availability**	It is always available in all platforms through website
NFR-6	**Scalability**	It is convenient for the user to use the application and also this app has been considered user friendly
NFR-7	**Learnability**	A new user should be able to use the recommender engine with ease. In case of any queries there must be some help provided to solve their queries

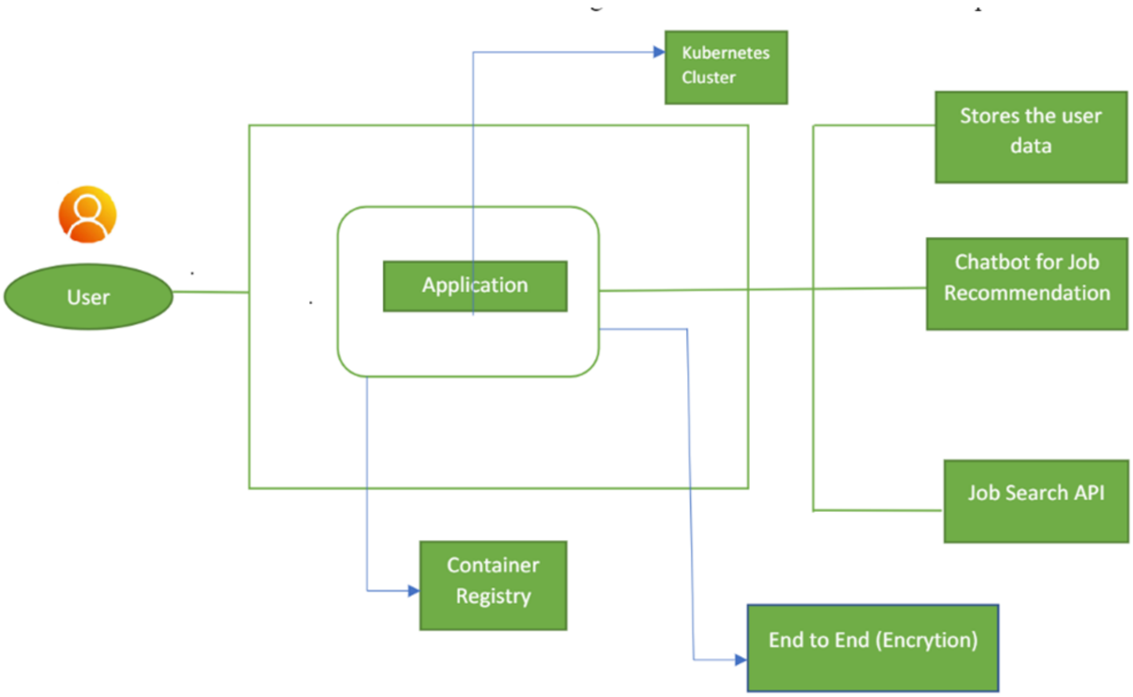
5. PROJECT DESIGN :

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.

5.2 SOLUTION ARCHITECTURE:



5.3 TECHNICAL ARCHITECTURE:



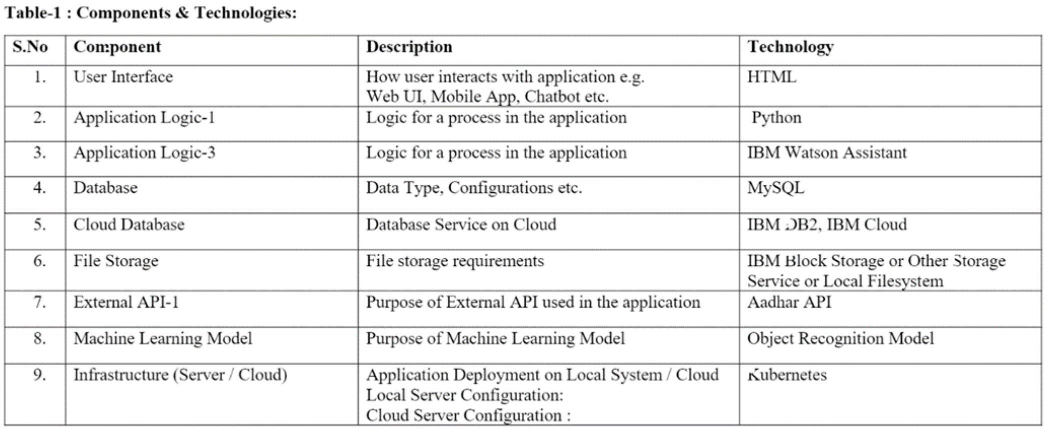


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
4.	Database	Data Type, Configurations etc.	MySQL
5.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud
6.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
7.	External API-1	Purpose of External API used in the application	Aadhar API
8.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Kubernetes

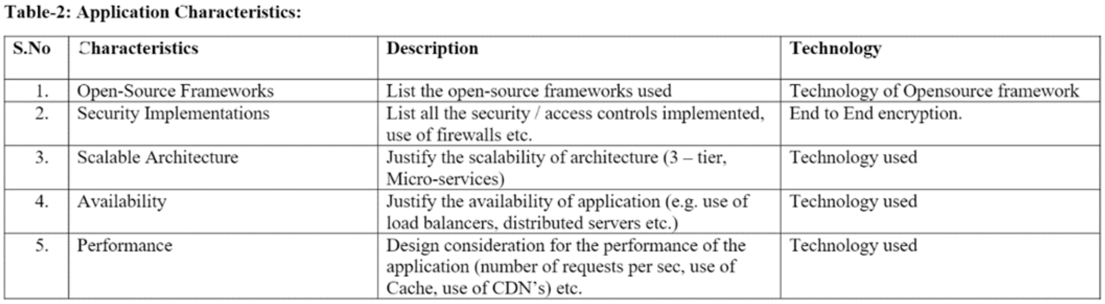
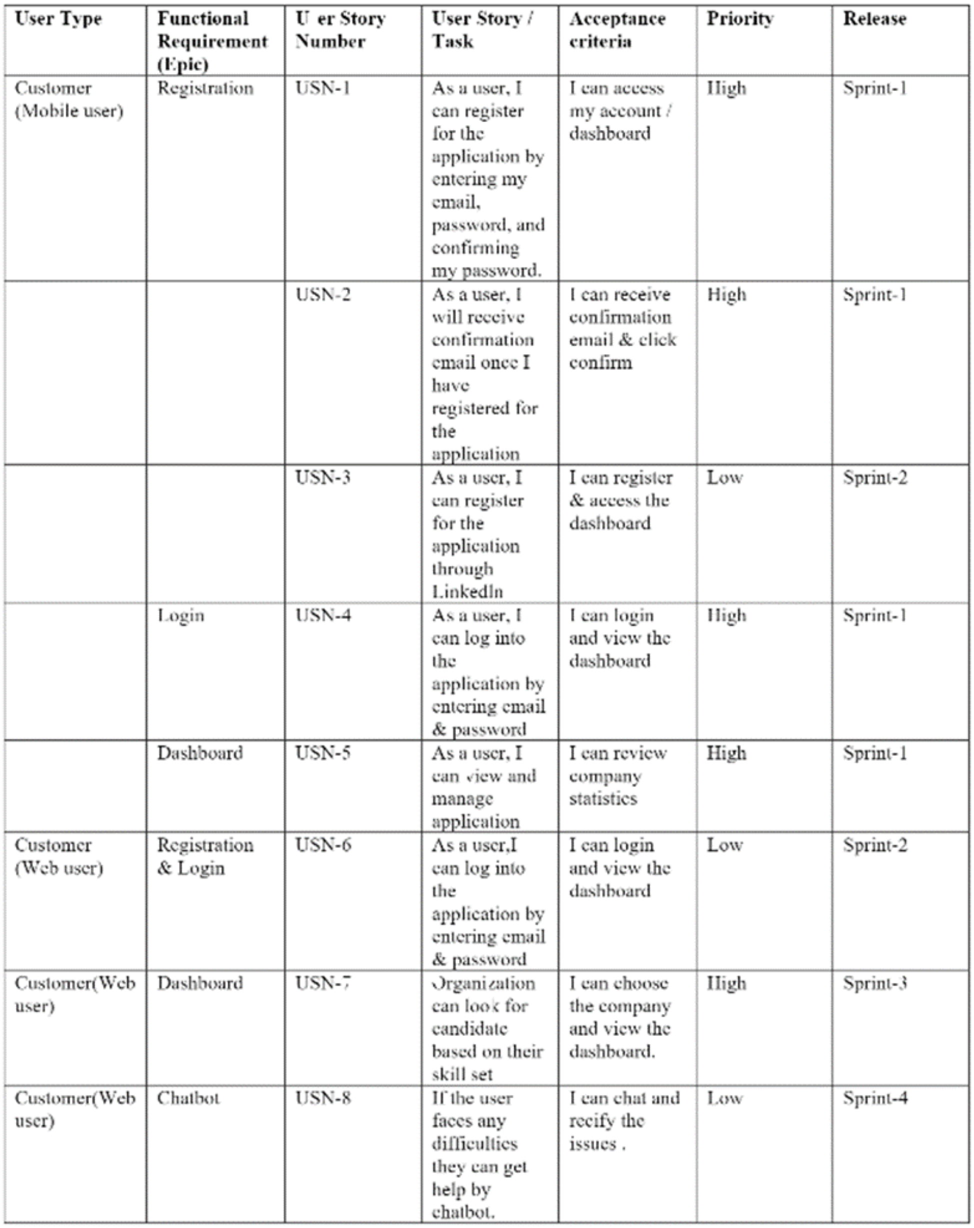


Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	End to End encryption.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

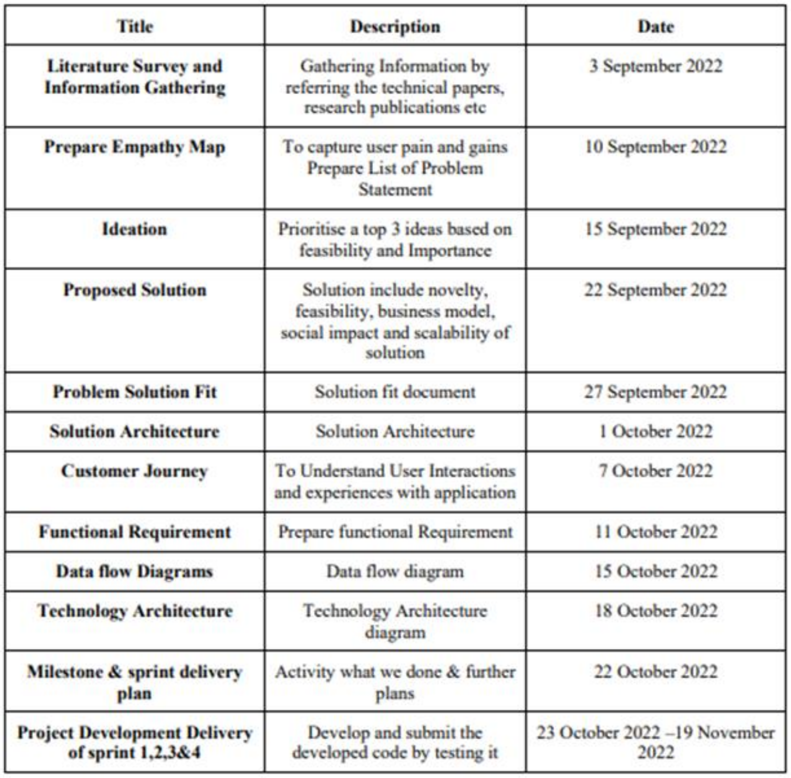
5.4 USER STORIES:



User Type	Functional Requirement (Epic)	U er Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through LinkedIn	I can register & access the dashboard	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login and view the dashboard	High	Sprint-1
	Dashboard	USN-5	As a user, I can view and manage application	I can review company statistics	High	Sprint-1
Customer (Web user)	Registration & Login	USN-6	As a user,I can log into the application by entering email & password	I can login and view the dashboard	Low	Sprint-2
Customer(Web user)	Dashboard	USN-7	Organi zation can look for candidate based on their skill set	I can choose the company and view the dashboard.	High	Sprint-3
Customer(Web user)	Chatbot	USN-8	If the user faces any difficulties they can get help by chatbot.	I can chat and recify the issues .	Low	Sprint-4

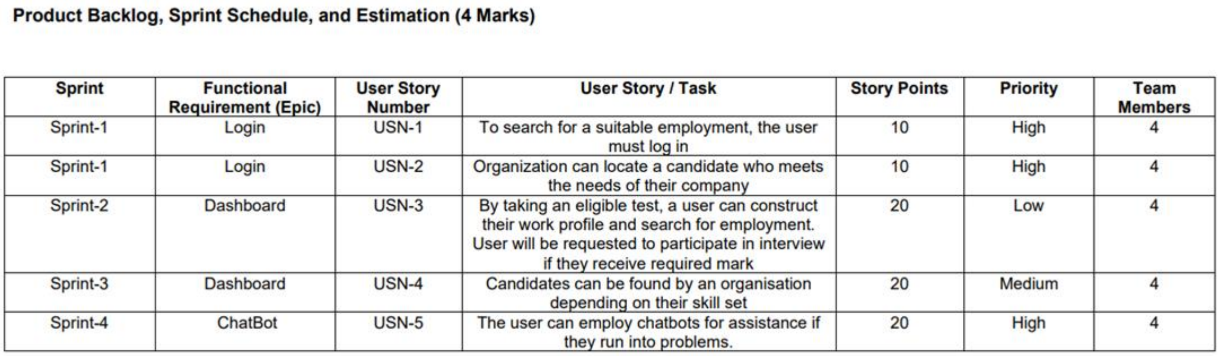
6.PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation



Title	Description	Date
Literature Survey and Information Gathering	Gathering Information by referring the technical papers, research publications etc	3 September 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	10 September 2022
Ideation	Prioritise a top 3 ideas based on feasibility and Importance	15 September 2022
Proposed Solution	Solution include novelty, feasibility, business model, social impact and scalability of solution	22 September 2022
Problem Solution Fit	Solution fit document	27 September 2022
Solution Architecture	Solution Architecture	1 October 2022
Customer Journey	To Understand User Interactions and experiences with application	7 October 2022
Functional Requirement	Prepare functional Requirement	11 October 2022
Data flow Diagrams	Data flow diagram	15 October 2022
Technology Architecture	Technology Architecture diagram	18 October 2022
Milestone & sprint delivery plan	Activity what we done & further plans	22 October 2022
Project Development Delivery of sprint 1,2,3&4	Develop and submit the developed code by testing it	23 October 2022 –19 November 2022

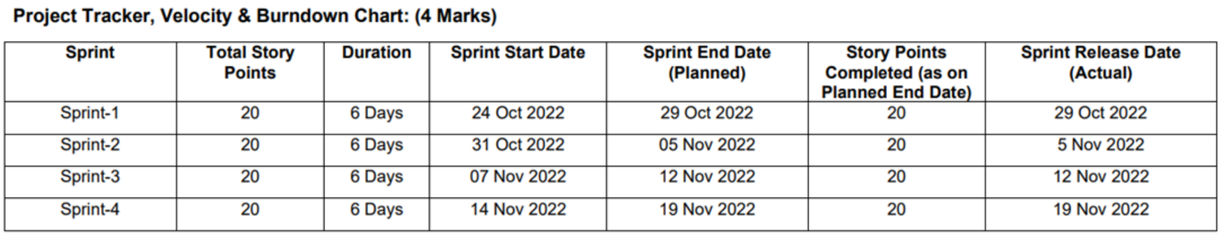
6.2 Sprint Delivery Schedule



Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	To search for a suitable employment, the user must log in	10	High	4
Sprint-1	Login	USN-2	Organization can locate a candidate who meets the needs of their company	10	High	4
Sprint-2	Dashboard	USN-3	By taking an eligible test, a user can construct their work profile and search for employment. User will be requested to participate in interview if they receive required mark	20	Low	4
Sprint-3	Dashboard	USN-4	Candidates can be found by an organisation depending on their skill set	20	Medium	4
Sprint-4	ChatBot	USN-5	The user can employ chatbots for assistance if they run into problems.	20	High	4

Project Tracker, Velocity & Burn down Chart:



Project Tracker, Velocity & Burndown Chart: (4 Marks)

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	5 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

Velocity:

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

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

7. CODING & SOLUTIONING

1. Self-service On-Demand

This is one of the most essential and significant characteristics of cloud computing. This means that cloud computing enables clients to regularly monitor the abilities, allotted network storage, and server uptime. Therefore, it is one of the most fundamental features of cloud computing that helps clients control various computing abilities as per their requirements.

2. Resources Pooling

This is also a fundamental characteristic of cloud computing. Pooling resources means that a cloud service provider can distribute resources for more than one client and provide them with different services according to their needs. Resource Pooling is a multi-client plan useful for data storing, bandwidth services and data processing services. The provider administers the data stored in real-time without conflicting with the need for data.

3. Easy Maintenance

This is one of the best cloud characteristics. Cloud servers are easy to maintain with low to almost zero downtime. Cloud Computing powered resources undergo several updates frequently to optimize their capabilities and potential. The updates are more viable with the devices and perform quicker than the previous versions.

4. Economical

This kind of service is economical as it efficiently reduces IT costs and data storage expenditure. Moreover, most cloud computing services are free. Even if there are paid plans, it's only to expand storage capacity, and these costs are often very nominal. This is a massive advantage of using cloud computing services.

5. Rapid Elasticity and Scalability

The best part of using cloud storage is that it can easily handle all the workload and data load concerning storage. Furthermore, as it is fully automated, businesses and organizations can save heavily on manual labor and technical staffing as cloud services are elastic, scalable and automated. This is one of the significant advantages of using cloud services.

API Features:

1. HTTPSISSL certificates Programming cheat sheets

2. Try for free: Red Hat Learning Subscription

3. eBook: An introduction to programming with Bash Bash Shell Scripting Cheat Sheet

4. eBook: Modernizing Enterprise Java

The gold standard for the web is HTTPS using SSL certificates, and Let's Encrypt can help you achieve this. It is a free, automated, and open certificate authority from the non-profit Internet Security Research Group (ISRG).

2. Cross-origin resource sharing

CORS is a browser-specific security policy preflight check. If your APIserver is not in the same domain as the requesting client's domain, you will need to deal with CORS. For example, if your server is running onapi.domain-a.com and gets a client request from domain-b.com, Coarsens an HTTP precheck request to see if your API service will accept client-side requests from the client's domain,

3. Authentication and JSON Web Tokens

There are several approaches to validate an authenticated user in your API, but one of the best ways is to use JSON Web Tokens (JWT). These tokens are signed using various types of well-known cryptographic libraries. When a client logs in, an identity-management service provides the client with a JWT. The client can then use this token to make requests to the API. The API has access to a public key or a secret that it uses to verify the token. There are several libraries available to help verify tokens, including honeytoken. For more information about JWT and the libraries that support it in every language, check out JWT.io.

4. Authorizations and scopes

Authentication (or identity verification) is important, but so isauthorization, i.e., does the verified client have the privilege to execute this request? This is where scopes are valuable. When the client authenticates with the identity management server and a JWT token is created, having the identity management service provide the scopes for the given authenticated client can enable the API service to determine if this verified client request can be performed without having to perform an additional costly lookup to an access control list.

7.2 FEATURE 2 Docker Features:

1. Faster and Easier configuration:

It is one of the key features of Docker that helps you in configuring the system in a faster and easier manner. Due to this feature, codes can be deployed in less time and with fewer efforts. The infrastructure is not linked with the environment of the application as Docker is used with a wide variety of environments.

2. Application isolation:

Docker provides containers that are used to run applications in an isolated environment. Since each container is independent, Docker can execute any kind of application.

1. Increase in productivity:

It helps in increasing productivity by easing up the technical configuration and rapidly deploying applications. Moreover, it not only provides an isolated environment to execute applications, but it reduces the resources as well.

2. Swarm:

Swarm is a clustering and scheduling tool for Docker containers. At the front end, it uses the Docker API, which helps us to use various tools to control it. It is a self-organizing group of engines that enables pluggable backends.

3. Services:

Services is a list of tasks that specifies the state of a container inside a cluster. Each task in the Services lists one instance of a container that should be running, while Swarm schedules them across the nodes.

Kubernetes Features:

1. Auto-scaling. Automatically scale containerized applications and their resources up or down based on usage
2. Lifecycle management. Automate deployments and updates with the ability to:

a. Rollback to previous versions

b. Pause and continue a deployment

3. Declarative model. Declare the desired state, and K8s works in the background to maintain that state and recover from any failures.
4. Resilience and self-healing. Auto placement, auto restart, auto replication and auto scaling provide application self-healing

5. Persistent storage. Ability to mount and add storage dynamically

6. Load balancing. Kubernetes supports a variety of internal and external load balancing options to address diverse needs.

7.2 DATABASE SCHEMA

Username: Varchar (32)

Email: Varchar (32)

Phone Number: Varchar (32)

Password: Varchar (32)

Pin: Varchar

8. TESTING

1. Verify that after registration users are navigated to login page
2. Verify the Ul elements in Login/Signup popup
3. Verify user can log into application with Valid credentials
4. Verify that categories of news are shown in homepage
5. Verify that news is displayed in homepage
6. Verify that when clicked on news it is redirected to correct page

8.2 USER ACCEPTANCE TESTING

The purpose of this documents to briefly explain the test coverage and open issues of the Recommender Application project at the time of the release User Acceptance Testing

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

1. Defect Analysis

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

2. Test Case Analysis

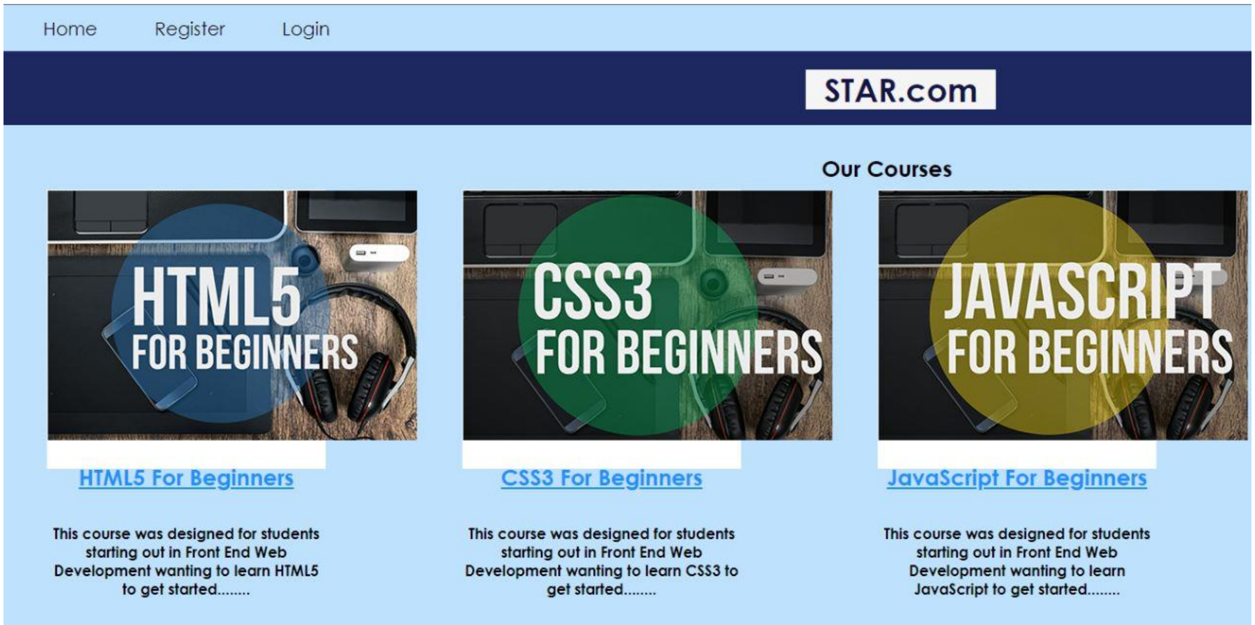
This report shows the number of test cases that have passed, failed, and untested

9. RESULT

9.1 PERFORMANCE METRICS

The application performance index, or Apex score, has become an industry standard for tracking the relative performance of an application. It works by specifying a goal for how long a specific web request or transaction should take.

Those transactions are then bucketed into satisfied (fast), tolerating (sluggish), too slow, and failed requests. A simple math formula is then applied to provide a score from 0 to 1.



Our Courses

HTML5 For Beginners

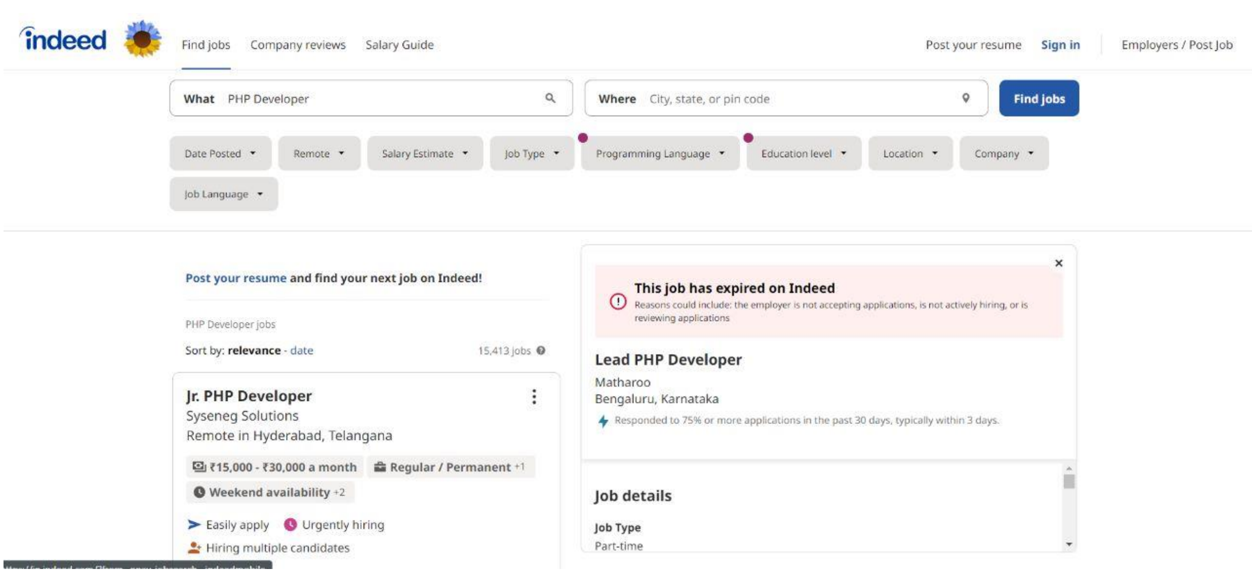
This course was designed for students starting out in Front End Web Development wanting to learn HTML5 to get started........

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........



What PHP Developer

Where City, state, or pin code

Post your resume and find your next job on Indeed!

PHP Developer jobs

Sort by: relevance - date 15,413 jobs

Jr. PHP Developer
Syseneg Solutions
Remote in Hyderabad, Telangana

₹15,000 - ₹30,000 a month Regular / Permanent +1

Weekend availability +2

Easily apply Urgently hiring

Hiring multiple candidates

This job has expired on Indeed

Reasons could include: the employer is not accepting applications, is not actively hiring, or is reviewing applications

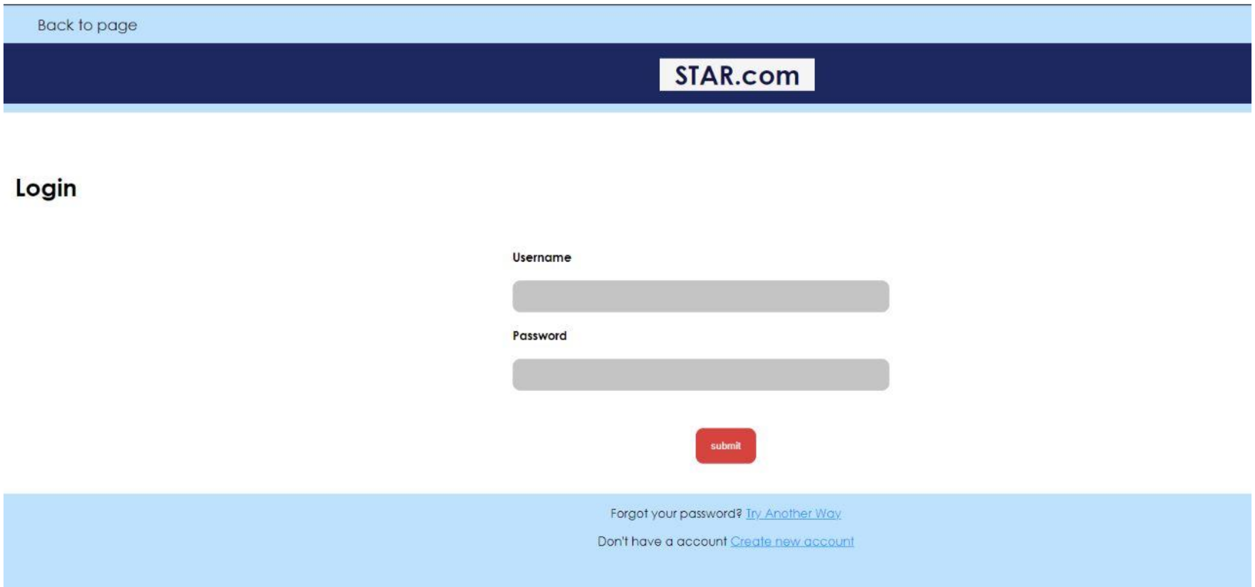
Lead PHP Developer

Matharoo
Bengaluru, Karnataka

Responded to 75% or more applications in the past 30 days, typically within 3 days.

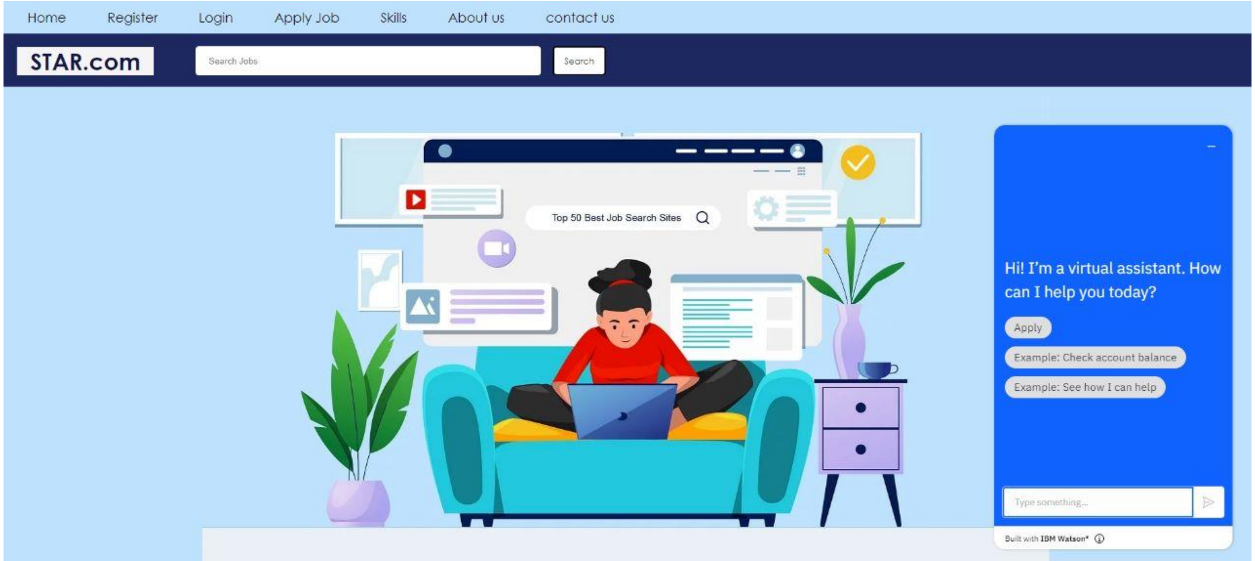
Job details

Job Type
Part-time

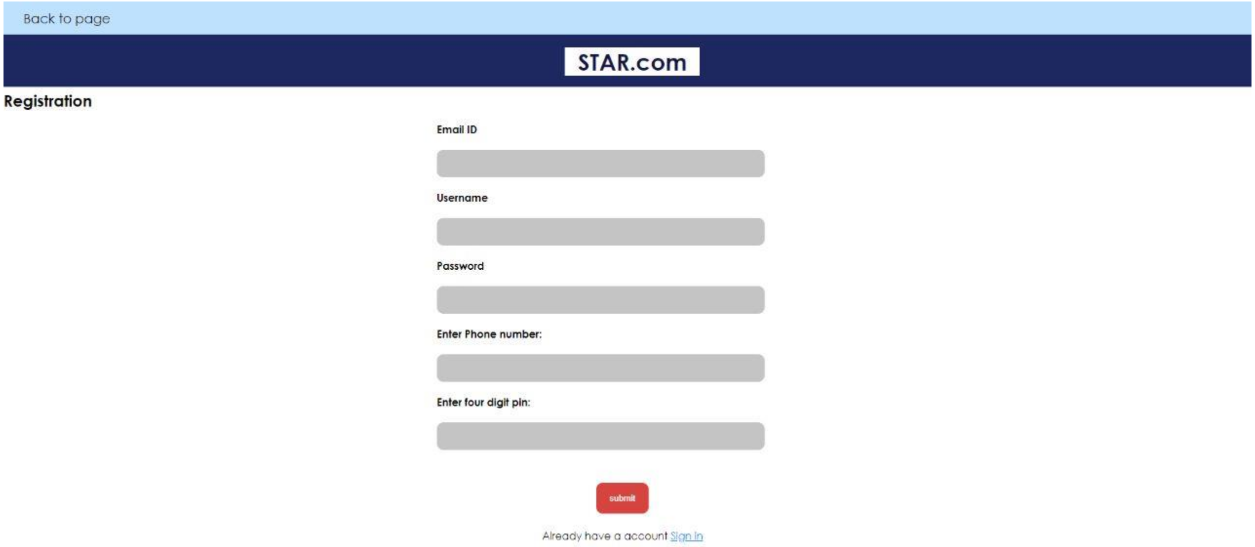


Forgot your password? Try Another Way

Don't have a account Create new account



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


Back to page

STAR.com

Registration

Email ID

Username

Password

Enter Phone number:

Enter four digit pin:

submit

Already have a account Sign in

9.2ADVANTAGES AND DISADVANTAGES

1. Employment Opportunities:

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

2. Easy Job Applications:

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

3. Initiate Connections:

Apart from receiving a job offer, the connections you establish on your profile help you in the long run. For instance, you may start by connecting with your school and college friends and eventually shift to your colleagues. An alumnus from your university is good connections to have. Having an illustrious list of connections speaks to your strong profile. Having a connection who is working at your dream company can be your pathway to the same. Initiating connections will allow you to analyze industry trends and be at the top of the game.

4. Endorsement and Connections

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

Disadvantages:

1. Risk of identity theft

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

2. Incomplete profile challenge

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

3. Tons of spam messages

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

4. Premium package can be expensive

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

11. CONCLUSION

We proposed an application for job recommendation task. This application facilitates the opportunities 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. 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.

12. FUTURE SCOPE

The future is and will remain unknown to us , but fact is there is a issue of unemployment in the world of job market, the job portals will exist and will grow in proportionate with demand

13. APPENDIX

Source Code:

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(20, 20, 70);background-color: whitesmoke;margin-left: 30px;padding-left: 20px;padding-right: 20px;">SKYS.com</span></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>
```

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('home')}}">Back to page</a></li>

            </ul>

        </nav>

    </header>
```

```
        <section id="sss1">

            <center><h1 class="two" ><span style="color:rgb(20, 20, 70);background-color:white;margin-left: 30px;padding-left: 20px;padding-right: 20px;">Star.com</span></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>

     </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>
```

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>Avatar</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;">Star.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>
```

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>Avatar.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;">Star.com</span></h1></center>

    </section>

    <br><br>

    <section id="jobs">



        <center><h1><b>APPLY JOBS</b></h1></center><br>

        <img style="position:absolute;top: 250px;border-radius: 20px;"src="{{url_for('static',filename='img/php.jpg')}}" alt=""width="370" height="200">

        <img style="position:absolute;top: 550px;border-radius: 20px;"src="{{url_for('static',filename='img/python.png')}}" alt=""width="370" height="200">

        <img style="position:absolute;top: 250px;left: 470px;border-radius: 20px;"src="{{url_for('static',filename='img/software.jpg')}}" alt=""width="370" height="200">

        <img style="position:absolute;top: 550px;left: 470px;border-radius: 20px;"src="{{url_for('static',filename='img/java.jpg')}}" alt=""width="370" height="200">

        <img style="position:absolute;top: 250px;left: 970px;border-radius: 20px;"src="{{url_for('static',filename='img/web.jpg')}}" alt=""width="370" height="200">

        <img style="position:absolute;top: 550px;left: 970px;border-radius: 20px;"src="{{url_for('static',filename='img/sql.jpg')}}" alt=""width="370" height="200">
        <h2 style="position:absolute;top: 450px;left:150px"><a href="https://in.Star.com/PHP-Developer-jobs?vjk=10cca9575b193c7d"><button>APPLY</button></a></h2>

        <h2 style="position:absolute;top: 450px;left:650px"><a href="https://in.Star.com/Software-Developer-jobs?vjk=b7da08f07cac87d5"><button>APPLY</button></a></h2>

        <h2 style="position:absolute;top: 450px;left:1150px"><a href="https://in.Star.com/Web-Developer-jobs?vjk=b81e49165da51eeb"><button>APPLY</button></a></h2>

        <h2 style="position:absolute;top: 750px;left:1150px"><a href="https://in.Star.com/SQL-Developer-jobs?vjk=ac86b15908022123"><button>APPLY</button></a></h2>

        <h2 style="position:absolute;top: 750px;left:650px"><a href="https://in.Star.com/Java-Developer-jobs?vjk=da306a665e00eb30"><button>APPLY</button></a></h2>

        <h2 style="position:absolute;top: 750px;left:150px"><a href="https://in.Star.com/Python-Developer-jobs?vjk=fa7b9bd250044569"><button>APPLY</button></a></h2>

    </section>

</body>

</html>
```

Forgot.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;">Star</span>.com</h1></center>
        </section>
        <br><br>

        <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>
```

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>Star.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;">Star.com</span></h1></center>
    </section>
    <br><br>
    <center><h1 class="h1lt">Contact us</h1></center>

    <form>

        <label class="form_label"><b>first name</b></label><br><br>
        <input class="form_input" type="text" name="firstname" value="">
 <br><br>

 <label class="form_label"><b>last name</b></label><br><br>
 <input class="form_input" type="text" name="lastname" value="">


 <br><br>


 <label class="form_label"><b>email</b></label><br><br>

 <input class="form_input" type="text" name="email" value="">

 <br>

 <br>


 <label class="form_label"><b>text area</b></label><br><br>

 <textarea class="form_input" type="text" name="text area" value=""></textarea>

 <br><br>

 <center><input type="submit" class="submitbtn"value="submit" /></center>

</form>


</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>Star.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;">Star.com</span></h1></center>


    </section>
    <br>

    <br>

    <center><h1><b>ABOUT US

    </b></h1></center>

    <P> <b> <h3> Hello Everyone! We did this project for IBM Nalaiyathiran. Our
team consists of four members. Our Project is: Skill/Job Recommender Application.
This project recommends for Job seekers and also for skilled person. We heartfully
Thank IBM for providing their free resources and we gained lot of knowledge from
learning new technologies.  </h3></b></P>

<h1>TEAM : </h1>

<H3>TEAM LEADER   : Swetha S</H3>

<H3>TEAM MEMBER 1 : Selvalakshmi S</H3>

<H3>TEAM MEMBER 2 : Roshan S</H3>

<H3>TEAM MEMBER 3 : Pavithra V</H3>

</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__,template_folder='Templates')
app.secret_key='a'
try:
    conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=b0aebb68-94fa-46ec-a1fc-
1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud;PORT=31249;SECURITY=SS
L;SSLServerCertificat=DigiCertGlobalRootCA.crt;UID=ymd42279;PWD=gygu16zNcomD2Bxj",
'','')
except:
    print("Unable to connect: ",ibm_db.conn_error())

@app.route("/")
def dash():
    return render_template('home.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,host="0.0.0.0")
```

Docker File:

```
FROM python:3.6

WORKDIR /app

ADD . /app

COPY requirements.txt /app

RUN python3 -m pip install -r requirements.txt

RUN python3 -m pip install ibm_db

EXPOSE 5000

CMD ["python","app.py"]
```

Requirements File:

flask

ibm_db

sendgrid

Requests

Service.yaml:

```
apiVersion: v1
kind: Service
metadata:
  name: flask-node-deployment
spec:
  ports:
  - port: 5000
    targetPort: 5000
  selector:
    app: flasknode
```

Deploy.yaml:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: flask-node-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: flasknode
  template:
    metadata:
      labels:
        app: flasknode
    spec:
      containers:
      - name: flasknode
        image: us.icr.io/job-skill/job-skill
        imagePullPolicy: Always
        ports:
        - containerPort: 5000
```

GitHub & Project Demo Link:

Github Link:

https://github.com/IBM-EPBL/IBM-Project-19517-1668782009

Demo Link:

https://loom.com/share/e38570155e6a416eae1489c4aa4aba17