SKILL/JOB RECOMMENDER APPLICATION

PROJECT REPORT

Submitted by

OBULIPURUSOTHAMAN K [19CS094]

SELVAKUMAR R [19CS110]

SARAVANAN G [19CS109]

SOWNDHAR S [19CS113]

In partial fulfillment for the award of the degree

Of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

MUTHAYAMMAL ENGINEERING COLLEGE (AUTONOMOUS)

RASIPURAM - 637 408

ANNA UNIVERSITY::CHENNAI- 600 025

JUNE 2023

Date	19 November 2022
Team ID	PNT2022TMID18790
Project Name	Project- Skill and Job Recommender
Team Leader/ Team Members	Obulipurusothaman K Selvakumar R Saravanan G Sowndhar S

ABSTRACT

In proposed work recommender systems have gained rural graduates popularity in recent years because they inefficiently alleviate information overload by delivering individualized job suggestions. Although they are from rural place so this platform provides several ways and practices for using job recommender systems in the literature, the majority of them fall short of recommending positions that are correctly matched to the profiles of rural graduate job searchers is the platform its very useful to them and through that they can search the job very easily according to their skills. In the Skill/Job Recommendation Application its has a various number of job opportunities for a various skills and the process of accessing the application is very useful 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.

Project Report

LIST OF FIGURES

LIST OF ABBREVIATIONS

- API-APPLICATION PROGRAMMING INTERFACE
- DB2-DATABASE 2
- CLI-COMMAND LINE INTERFACE
- HTML-HYPER TEXT MARKUP LANGUAGE
- JS-JAVA SCRIPT
- K8S-KUBERNETES
- DC-DOCKER CONTAINER
- UI-USER INTERFACE

CHAPTER 1

INTRODUCTION

CLOUD COMPUTING

Cloud computing, sometimes referred to simply as “cloud,” is the use of computing resources — servers, database management, data storage, networking, software applications, and special capabilities such as blockchain and artificial intelligence (AI) — over the internet, as opposed to owning and operating those resources yourself, on premises. Compared to traditional IT, cloud computing offers organizations a host of benefits: the cost-effectiveness of paying for only the resources you use; faster time to market for mission-critical applications and services; the ability to scale easily, affordably and — with the right cloud provider — globally; and much more (see “What are the benefits of cloud computing?” below). And many organizations are seeing additional benefits from combining public cloud services purchased from a cloud services provider with private cloud infrastructure they operate themselves to deliver sensitive applications or data to customers, partners and employees.

Increasingly, “cloud computing” is becoming synonymous with “computing.” For example, in a 2019 survey of nearly 800 companies, 94% were using some form of cloud computing (link resides outside IBM). Many businesses are still in the first stages of their cloud journey, having migrated or deployed about 20% of their applications to the cloud, and are working out the unique security, compliance and geographic implications of moving their remaining mission-critical applications. But move they will: Industry analyst Gartner predicts that more than half of companies using cloud today will move to an all-cloud infrastructure by next year (2021) (link resides outside IBM).

SKILL/JOB RECOMMENDER

In proposed work recommender systems have gained rural graduates popularity in recent years because they inefficiently alleviate information overload by delivering individualised job suggestions. Although they are from rural place so this platform provides several ways and practices for using job recommender systems in the literature, the majority of them fall short of recommending positions that are correctly matched to the profiles of rural graduate job searchers is the platform its very useful to them and through that they can search the job very easily according to their skills

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. Commonly, a job seeker has two ways to search a job using these sites: 1) doing a query based on keywords related to the job vacancy that he/she is looking for, or 2) creating and/or updating a professional profile containing data related to his/her education, professional experience, professional skills and other, and receive personalized job recommendations based on this data. Sites providing support to the former case are more popular and have a simpler structure; however, their recommendations are less accurate than those of the sites using profile data. Personalized job recommendation sites implemented a variety of types of recommender systems, such as content-based filtering, collaborative filtering, knowledge-based and hybrid approaches.

1.1 PROJECT OVERVIEW

Overview of **SKILL/JOB RECOMMENDER APPLICATION** Simply put, it is **a system that gives us recommendations based on the data that it has**

collected from us, and other users like us, over a course of time. These systems today, work in areas like movies, music, news, research articles, search queries, restaurants, hashtags, and more. More over its very useful to the rural people as well as the rural are gradutes.

1.2 PURPOSE

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 e-recruiting functionality, many recommender system approaches have been proposed. This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

- By analyzing the customer's present site use and his previous browsing history, a recommendation engine can deliver appropriate product suggestions as he stores.
- The data is gathered in real-time so the software can respond as his shopping habits change.
- Providing guides is an integral part of a personalization system.
- Providing the client precise and up to the minute reporting permits him to make solid choices about his website and the direction of a project.
- An experienced carrier can provide suggestions on ways to utilize the data gathered and reported to the customer

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

EFFICIENT AND SCALABLE JOB RECOMMENDER SYSTEM USING COLLABORATIVE FILTERING

AUTHORS: Ravita Mishra (&) Sheetal Rathi

ALGORITHM: Collaborative Filtering

Now-a-day social media is very common platform to share the data and day today's activities. With the enormous use of various internet sources likes, mobile phone and smart devices, Internet users can receive huge information about shopping and social activity of user and online learning. If the data volume and variety increases tremendously, then individual user faces various problem of excessive information, it causes problem to make the correct decisions. This framework is called as information overload. To resolve users' information overload problem, a new technique recommender system comes in pictures. Recommender system can solve various problems by effectively finding users' probable requirements and elect fascinating items from a vast amount of applicant information. Recommender systems are mainly categorized into three main forms, i.e., content-based (CB), collaborative filtering (CF) and hybrid recommender system is combination of both resolve the drawback of content and collaborative filtering.

LIMITATIONS

- Data Sparsity and cold-start problem. Data sparsity is seen as a key disadvantage of collaborative filtering
- Cold-star
- Scalability

ENHANCED JOB RECOMMENDATION SYSTEM

AUTHORS: Shivraj Hulbatte, Amit Wabale, Suraj Patil, Nikhilkumar Sathe

ALGORITHM: Expectation Maximization (EM)

The increase in usage of Internet has heightened the need for online job hunting. According to Job site's report 2014, 68% of online job seekers are college graduates or post graduates. The key problem is that most of the job-hunting websites just display the recruitment information to website viewers. Students have to go through all the information to find the jobs they want to apply. The whole procedure is tedious and inefficient. We need an easy job recommendation system where everyone will have a fair and square chance. This saves a lot of potential time and money both, on the industrial as well as the job seeker's side. Moreover, as the candidate gets a fair chance to prove his talent in the real world it is a lot more efficient system. The basic agenda of every algorithm used in today's world, be it a traditional algorithm or a hybrid algorithm, is to provide a suitable job that the user actually seeks and wishes for. Recently, job recommendation has attracted a lot of research attention and has played an important role on the online recruiting website.

LIMITATIONS

- Significant investments required.
- The complex onboarding process.
- Inability to capture changes in user behavior.

2.1 EXISTING SYSTEM

ALGORITHM: Collaborative Filtering

Collaborative filtering filters information by using the interactions and data

collected by the system from other users. It's based on the idea that people who agreed in their evaluation of certain items are likely to agree again in the future.

The concept is simple: when we want to find a new movie to watch we'll often ask our friends for recommendations. Naturally, we have greater trust in the recommendations from friends who share tastes similar to our own.

Most collaborative filtering systems apply the so-called similarity index-based technique. In the neighborhood-based approach, a number of users are selected based on their similarity to the active user. Inference for the active user is made by calculating a weighted average of the ratings of the selected users.

Collaborative-filtering systems focus on the relationship between users and items. The similarity of items is determined by the similarity of the ratings of those items by the users who have rated both items.

LIMITATIONS

- Pearson similarity
- Jaccard similarity
- Spearman rank correlation
- Mean squared differences
- Proximity–impact–popularity similarity

ALGORITHM: Expectation Maximization (EM)

In the real-world applications of machine learning, it is very common that there are many relevant features available for learning but only a small subset of them are observable. So, for the variables which are sometimes observable and sometimes not, then we can use the instances when that variable is visible is observed for the purpose of learning and then predict its value in the instances when it is not observable.

On the other hand, Expectation-Maximization algorithm can be used for the latent variables (variables that are not directly observable and are actually inferred from the

values of the other observed variables) too in order to predict their values with the condition that the general form of probability distribution governing those latent variables is known to us. This algorithm is actually at the base of many unsupervised clustering algorithms in the field of machine learning.

It was explained, proposed and given its name in a paper published in 1977 by Arthur Dempster, Nan Laird, and Donald Rubin. It is used to find the local maximum likelihood parameters of a statistical model in the cases where latent variables are involved and the data is missing or incomplete.

ALGORITHM:

Given a set of incomplete data, consider a set of starting parameters.

Expectation step (E – step): Using the observed available data of the dataset, estimate (guess) the values of the missing data.

Maximization step (M – step): Complete data generated after the expectation (E) step is used in order to update the parameters.

Repeat step 2 and step 3 until convergence.

The essence of Expectation-Maximization algorithm is to use the available observed data of the dataset to estimate the missing data and then using that data to update the values of the parameters. Let us understand the EM algorithm in detail.

- Initially, a set of initial values of the parameters are considered. A set of incomplete observed data is given to the system with the assumption that the observed data comes from a specific model.
- The next step is known as “Expectation” – step or E-step. In this step, we use the observed data in order to estimate or guess the values of the missing or incomplete data. It is basically used to update the variables.

- The next step is known as “Maximization”-step or M-step. In this step, we use the complete data generated in the preceding “Expectation” – step in order to update the values of the parameters. It is basically used to update the hypothesis.
- Now, in the fourth step, it is checked whether the values are converging or not, if yes, then stop otherwise repeat step-2 and step-3 i.e. “Expectation” – step and “Maximization” – step until the convergence occurs.

LIMITATIONS

- It can be used to fill the missing data in a sample.
- It can be used as the basis of unsupervised learning of clusters.
- It can be used for the purpose of estimating the parameters of Hidden Markov Model (HMM).

2.2 REFERENCES

1. S. T. Al-Otaibi and M. Ykhlef, “A survey of job recommender systems,” International Journal of the Physical Sciences, vol. 7(29), pp. 5127-5142, July, 2012.
2. S. T. Zheng, W. X. Hong, N. Zhang and F. Yang, “Job recommender systems: a survey,” In Proceedings of the 7th International Conference on Computer Science & Education (ICCSE 2012), pp. 920-924, Melbourne, Australia, July, 2012.
3. M. Gao and Y. Q. Fu, “User-Weight Model for Item-based Recommendation Systems,” Journal of Software, vol. 7(9), pp. 2133-2140, 2012.
4. K. Yu, G. Guan and M. Zhou, “Resume information extraction with cascaded hybrid model,” In Proceedings of the 43rd Annual Meeting of the ACL, pp. 499-506, Ann Arbor, Michigan, June, 2005.
5. X. Yi, J. Allan and W. B. Croft, “Matching resumes and jobs based on relevance models,” In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 809-810, Amsterdam, The Netherlands, 2007.

6. I. Paparrizos, B. B. Cambazoglu and A. Gionis, “Machine learned job recommendation,” In Proceedings of the fifth ACM Conference on Recommender Systems, pp. 325-328, Chicago, USA, October, 2011.

7. J. S. Breese, D. Heckerman and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 42-52, 1998.

8. G. Adomavicius and A. Tuzhilin, “Toward the next generation of recommender systems: a survey of the stateof-the-art and possible extensions,” Knowledge and Data Engineering, IEEE Transactions on, vol. 17(6), pp. 734- 749, 2005.

9. H. W. Ye, “A Personalized Collaborative Filtering Recommendation Using Association Rules Mining and Self-Organizing Map,” Journal of Software, vol. 6(4), pp.732-739, 2011.

10. L. Hu, W. B. Wang, F. Wang, X. L. Zhang and K. Zhao, “The Design and Implementation of Composite Collaborative Filtering Algorithm for Personalized Recommendation,” Journal of Software, vol. 7(9), pp. 2040-2045, 2012.

11. F. Färber, T. Weitzel and T. Keim, “An automated recommendation approach to selection in personnel recruitment,” In Proceedings of the 2003 Americas Conference on Information Systems, pp. 2329-2339, Tampa, USA, 2003.

12. R. Burke, “Hybrid recommender systems: survey and experiments,” User Modeling and User-Adapted Interaction, vol. 12(4), pp. 331-370, 2002.

13. C. F. Chien and L. F. Chen, “Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry,” Expert Systems with Applications, vol. 34(1), pp. 280-290, 2008.

14. D. H. Lee and P. Brusilovsky, “Fighting information overflow with personalized comprehensive information access: a proactive job recommender,” In Proceedings of the Third International Conference on Autonomic and Autonomous Systems, Washington, DC, USA, 2007.

15. L. Pizzato, T. Rej, T. Chung, K. Yacef, I. Koprinska and J. Kay, “Reciprocal recommenders,” In Proceedings of 8th Workshop on Intelligent Techniques for Web Personalization and Recommender Systems, held in conjunction with the 18th International Conference on

User Modeling, Adaptation and Personalization (UMAP 2010), Hawaii, USA, June, 2010.

16. H. T. Yu, C. R. Liu and F. Z. Zhang, “Reciprocal recommendation algorithm for the field of recruitment,” Journal of Information & Computational Science, vol. 8(16), pp. 4061-4068, 2011.

17. J. Malinowski, T. Keim, O. Wendt and T. Weitzel, “Matching people and jobs: a bilateral recommendation approach,” In Proceedings of The 39th Hawaii International Conference on System Sciences, pp. 1-9, Hawaii, USA, 2006.

18. L. Li and T. Li, “MEET: a generalized framework for reciprocal recommender systems,” In Proceedings of the 21st ACM International Conference on Information and Knowledge Management, pp. 35-44, Hawaii, USA, 2012.

19. R. Burke, “Hybrid web recommender systems,” The Adaptive Web, vol. 4321, pp. 377-408, 2007.

20. T. Keim, “Extending the applicability of recommender systems: a multilayer framework for matching human resources,” In Proceedings of 40th Annual Hawaii International Conference on System Sciences, pp. 169-178, January, 2007.

21. M. Fazel-Zarandi and M. S. Fox, “Semantic matchmaking for job recruitment an ontolgy based hybrid approach,” In Proceedings of the 3rd International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web at the 8th International Semantic Web Conference, Washington D. C., USA, 2010.

2.3 PROBLEM STATEMENT DEFINITION

Customer Problem Statement :

Create a problem statement to understand your customer's point of view. The Customer Problem Statement template helps you focus on what matters to create experiences people will love.

A well-articulated customer problem statement allows you and your team to find the ideal solution for the challenges your customers face. Throughout the process, you'll also be able to empathize with your customers, which helps you better understand how they perceive

I am	Describe customer with 3-4 key characteristics - *who are they?*	Describe the customer and their attributes here
I'm trying to	*List their outcome or "job" the care about - what are they trying to achieve?*	List the thing they are trying to achieve here
but	Describe what problems or barriers stand in the way – *what bothers them most?*	Describe the problems or barriers that get in the way here
because	Enter the "root cause" of why the problem or barrier exists – *what needs to be solved?*	Describe the reason the problems or barriers exist
which makes me feel	Describe the emotions from the customer's point of view – *how does it impact them emotionally?*	Describe the emotions the result from experiencing the problems or barriers

your product or service.

Fig: 2.3.1

Reference: https://miro.com/templates/customer-problem-statement/

Example:

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Job Seeker	Find a suitable job.	Takes longtime to apply	It is difficult to apply for the job and no proper intimations about it.	Sad & Frustrated
PS-2	Job Seeker	Interact with AI	No proper guidance.	There is no proper chatbot for recommendation.	Frustrated
PS-3	Job Seeker/ Learner	Learn a new skill.	I don't have the skills whichare mentioned.	There this no proper recommendation for the skills to be learnt.	Sad
PS-4	Recruiter	Hire a Candidate	Take more time to hire.	There is no proper validation of the profile verification.	Sad
PS-5	Recruiter	Hire a Candidate	Invalid profile submission from various job seekers.	There is no proper constraint for the job profile.	Frustrated

CHAPTER 3

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

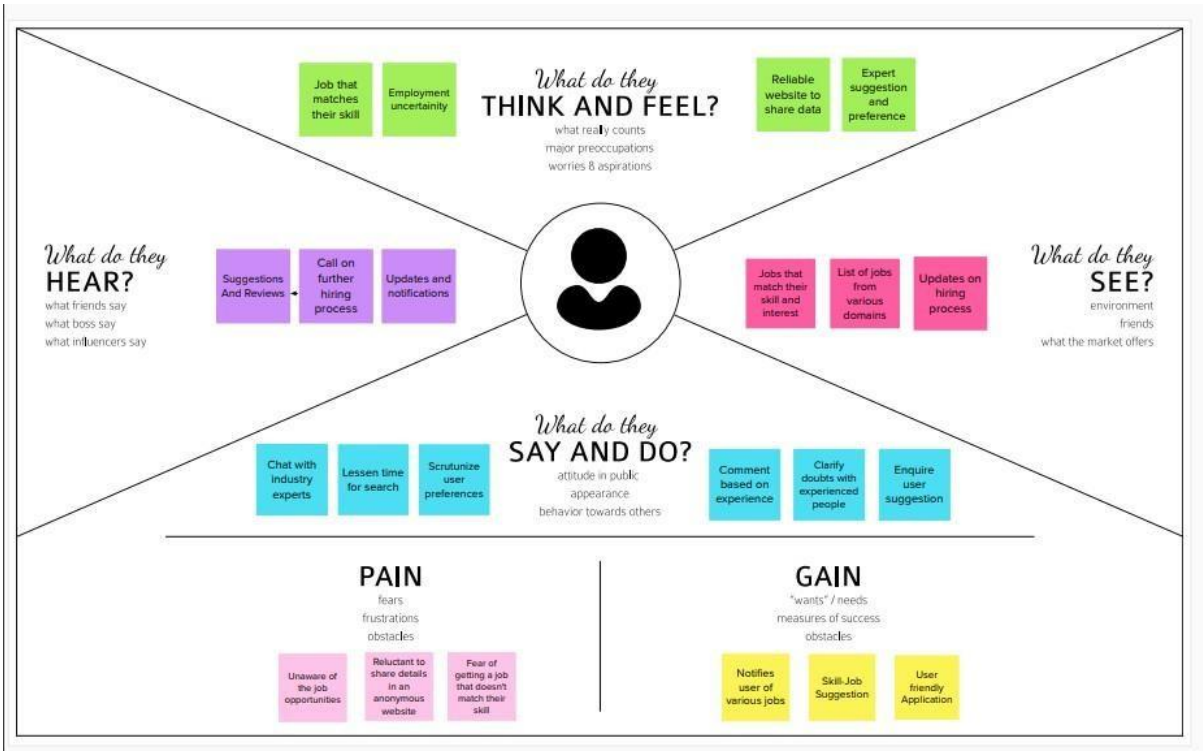


Fig: 3.1.1

An empathy map is a collaborative tool teams can use **to gain a deeper insight into their customers**. Much like a user persona, an empathy map can represent a group of users, such as a customer segment. The empathy map was originally created by Dave Gray and has gained much popularity within the agile community.

An Empathy Map consists of **four quadrants**. The four quadrants reflect four key traits, which the user demonstrated/possessed during the observation/research stage. The four quadrants refer to what the user: Said, Did, Thought, and Felt. It's fairly easy to determine what the user said and did.

3.2 IDEATION & BRAINSTORMING

Brainstorm & Idea Prioritization Template:

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

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

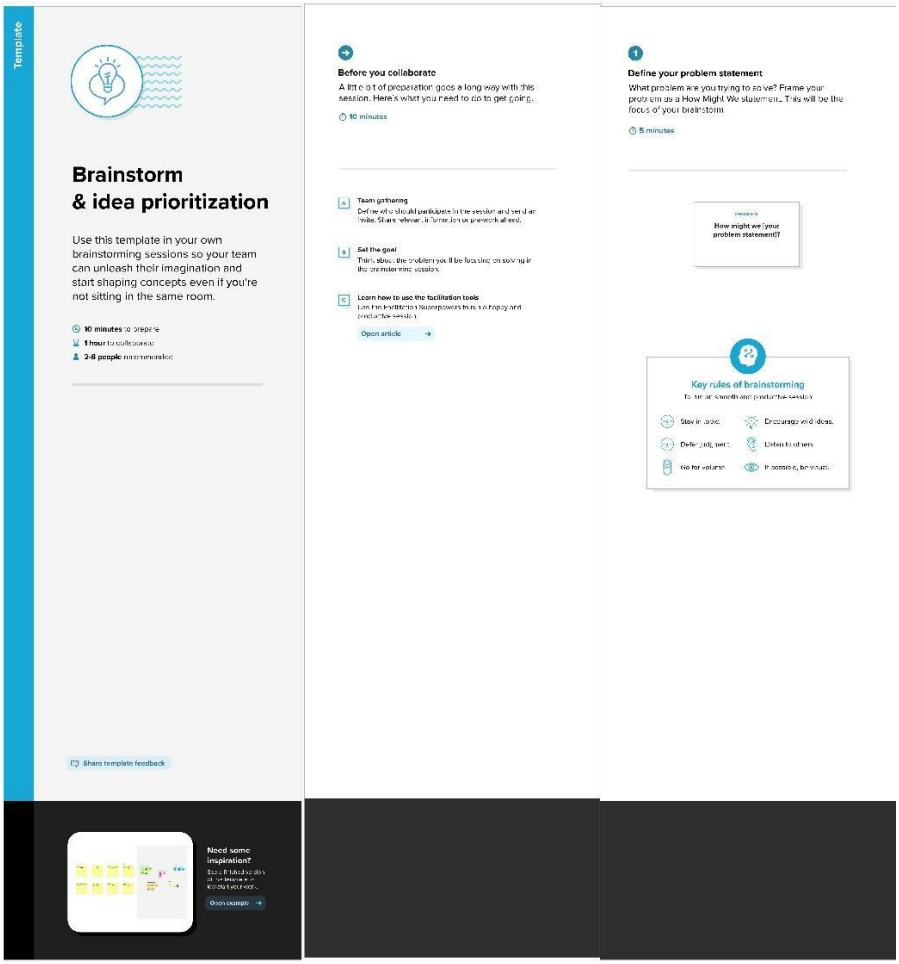


Fig: 3.2.1

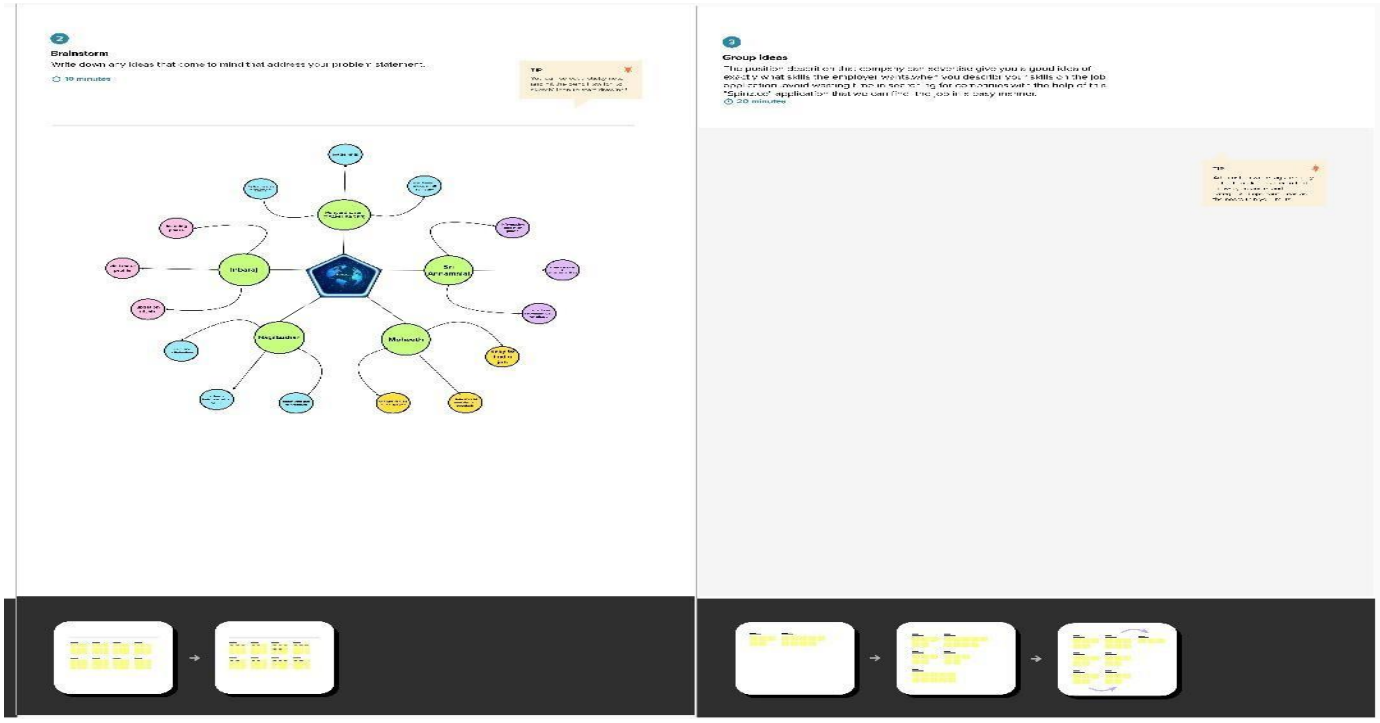


Fig: 3.2.2

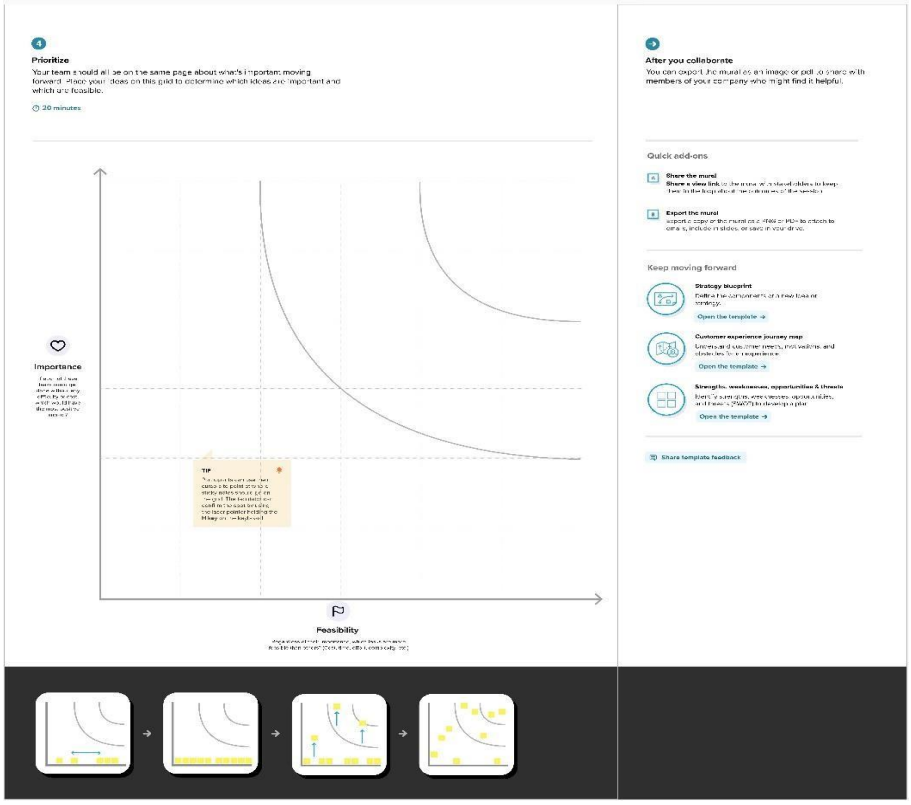


Fig: 3.2.3

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.

3.3 PROPOSED SOLUTION

PARAMETERS

PROBLEM IN JOB RECOMMENDATION

In Society many individuals face problems in job searching, many job seekers are unable to find their dream job. Many good technical persons are unable to land their dream job, and lose their hope. So wehave come up with a solution.

SOLUTION FOR THE PROBLEM

Our teammates, have designed a multi speciality software which helps job seekers to land intheir dream job according to their skills

NOVELTY

This software has designed to get Recommended job for the job seekers, the unique thing in this software is it has two types of account, one is vendor type account, another one is customer type, soThe job posted by the vendors can easily meet the customer

FEASIBILITY

The project is feasible and can implemented using flask framework, and the job API can be brought from third party service, and the software can be accessed from all over the world to meet job at all ends

BUSINESS MODEL

Apart from job recommendation, a revenueis important for a organization, so the required revenue can be brought up by third party ads like google ads

SOCIAL IMPACT

This software solves the social impacts likemaking all job seekers or individuals to meet the job that meets their criteria, so this can solve social issue on job finding

SCALABILITY

This software is based on SDLC, so the scalability of the software can be changed according to theneeds of customers in future

3.4 PROBLEM SOLUTION FIT

PROBLEMS AND SOLUTIONS

HOW CUSTOMERS MEET JOB?

The software uses two types of account, one is vendor type another is customer type, sothe job posted by vendors can be easily accessed by customers

HOW CUSTOMERS GET SUGGESTIONS?

As the profile created for customers, allthe experience and skill sets are gathered, so a special type of algorithm will provide suggestion about job that willmatch their profile

HOW CUSTOMERS CLARIFY THEIRPROBLEMS?

The software uses customer support facility and chatter bot, so any questions are clarified bothvendor and customer side

WHY JOB RECOMMENDATION APPLICATION?

Many individuals in society are without jobdue to many reasons so, we come up with online application it is easy to use and all individuals can apply for the job that fit for their skill

TIME AND MONEY?

As it is a online platform, the time andmoney can be saved, comparing to offline platform

CHAPTER 4

REQUIREMENT ANALYSIS

Functional Requirements:

Following are the functional requirements of the proposed solution.

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

Fig: 4.1

System Requirements:

8GB RAM,Intel Core i3,OS-Windows/Linux/MAC ,Laptop or Desktop

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.

Software Required:

Python, Flask ,Docker

PYTHON:

Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. This versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today. A survey conducted by industry analyst firm RedMonk found that it was the second-most popular programming language among developers in 2022

USES:

Python is commonly used for developing websites and software, task automation, data analysis, and data visualization. Since it's relatively easy to learn, Python has been adopted by many non-programmers such as accountants and scientists, for a variety of everyday tasks, like organizing finances.

"Writing programs is a very creative and rewarding activity," says University of Michigan and Coursera instructor Charles R Severance in his book *Python for Everybody*. "You can write programs for many reasons, ranging from making your living to solving a difficult data analysis problem to having fun to helping someone else solve a problem."

FLASK:

Flask Tutorial provides the basic and advanced concepts of the Python Flask framework. Our Flask tutorial is designed for beginners and professionals.Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO).

Flask is a web framework that provides libraries to build lightweight web applications in python. It is developed by **Armin Ronacher** who leads an international group of python enthusiasts (POCCO). It is based on WSGI toolkit and jinja2 template engine. Flask is considered as a micro framework.

USES:

Scalable

Size is everything, and Flask's status as a microframework means that you can use it to grow a tech project such as a web app incredibly quickly. If you want to make an app that starts small, but has the potential to grow quickly and in directions you haven't completely worked out yet, then it's an ideal choice. Its simplicity of use and few dependencies enable it to run smoothly even as it scales up and up.

Flexible

This is the core feature of Flask, and one of its biggest advantages. To paraphrase one of the principles of the Zen of Python, simplicity is better than complexity, because it can be easily rearranged and moved around.

Not only is this helpful in terms of allowing your project to move in another direction easily, it also makes sure that the structure won't collapse when a part is altered. The minimal nature of Flask and its aptitude for developing smaller web apps means that it's even more flexible than Django itself.

Easy to negotiate

Like Django, being able to find your way around easily is key for allowing web developers to concentrate on just coding quickly, without getting bogged down. At its core, the microframework is easy to understand for web developers, not just saving them time and effort but also giving them more control over their code and what is possible.

Lightweight

When we use this term in relation to a tool or framework, we're talking about the design of it—there are few constituent parts that need to be assembled and reassembled, and it doesn't rely on a large number of extensions to function. This design gives web developers a certain level of control.

Flask also supports modular programming, which is where its functionality can be split into several interchangeable modules. Each module acts as an independent building block, which can execute one part of the functionality. Together this means that the whole constituent parts of the structure are flexible, moveable, and testable on their own.

Documentation

Following the creator's own theory that "nice documentation design makes you actually write documentation," Flask users will find a healthy number of examples and tips arranged in a structured manner. This encourages developers to use the framework, as they can easily get introduced to the different aspects and capabilities of the tool.

DOCKER:

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run. Running Docker on AWS provides developers and admins a highly reliable, low-cost way to build, ship, and run distributed applications at any scale.

Docker works by providing a standard way to run your code. Docker is an operating system for containers. Similar to virtual machines virtualizes (removes the need to directly manage) server hardware, containers virtualize the operating system of a server. Docker is installed

on each server and provides simple commands you can use to build, start, or stop containers.

USES:

Docker helps to ensure that all the developers have access to all the necessary bits and pieces of the software they work on. So **if someone adds software dependencies, everyone has them when needed**. If it is just one developer, there is no such need. But even in this case, Docker may help, eg.Docker is a platform for packaging, deploying, and running applications. Docker applications run in containers that can be used on any system: a developer's laptop, systems on premises, or in the cloud. Containerization is a technology that's been around for a long time, but it's seen new life with Docker.

Non-Functional Requirements:

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

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

Fig: 4.2

CHAPTER 5

PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

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

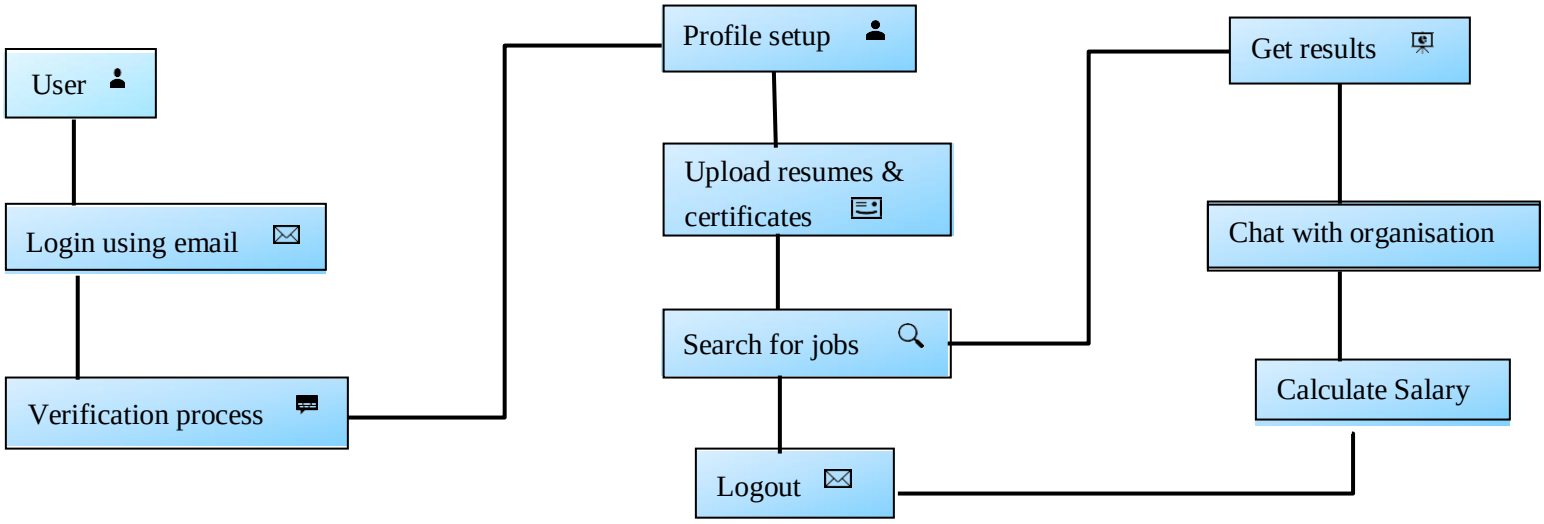


Fig: 5.1.1

5.2 SOLUTION & TECHNICAL ARCHITECTURE

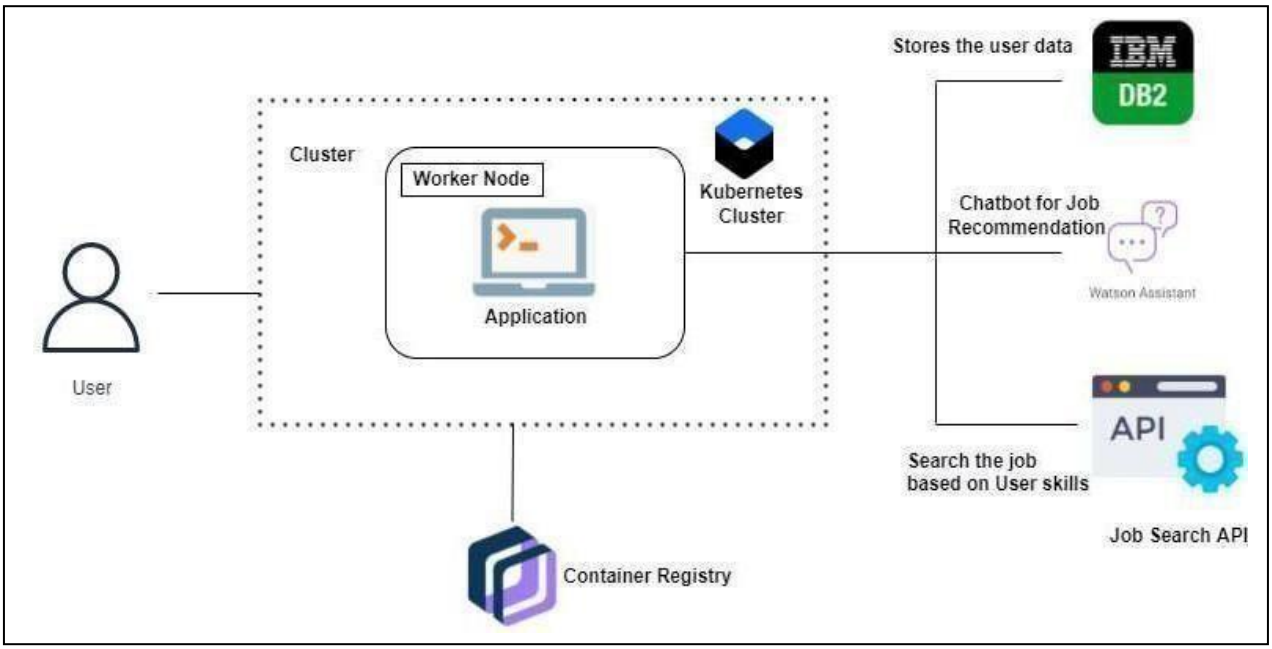


Fig: 5.1.2

IBM CLUSTER:

An IBM® i cluster is **a collection of one or more systems or logical partitions that work together as a single system**. Use this information to understand the elements and their relationship to each other. Cluster node. A cluster node is a IBM i system or logical partition that is a member of a cluster.

IBM KUBERNETES CLUSTER:

IBM Cloud® Kubernetes Service delivers powerful tools by combining Docker containers, the Kubernetes technology, an intuitive user experience, and built-in security and isolation to automate the deployment, operation, scaling, and monitoring of containerized apps in a cluster of compute hosts.

IBM CONTAINER REGISTRY:

IBM Cloud® Container Registry provides **a multi-tenant private image registry that you can use to store and share your container images with users in your IBM Cloud account**. The IBM Cloud console includes a brief Quick Start.

IBM DB2:

IBM Db2 is a family of data management products, including the Db2 relational database. The products feature AI-powered capabilities to **help you modernize the management of both structured and unstructured data across on-premises and multicloud environments**.

IBM WATSON:

IBM Watson is AI for business. Watson **helps organizations predict future outcomes, automate complex processes, and optimize employees' time**.

API:

API stands for **Application Programming Interface**. In the context of APIs, the word Application refers to any software with a distinct function. Interface can be thought of as a contract of service between two applications. This contract defines how the two communicate with each other using requests and responses.

VISUAL STUDIO IDE:

An integrated development environment (IDE) is a feature-rich program that supports many aspects of software development. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app. Over and above the standard editor and debugger that most IDEs provide, Visual Studio includes compilers, code completion tools, graphical designers, and

many more features to enhance the software development process.

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE.

5.3 USER STORIES

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirmingmy password.	I can access my account /dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation emailonce I have registered for the application	I can receive confirmationemail & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the applicationthrough Facebook	I can register & access the dashboard with FacebookLogin	Low	Sprint-2
		USN-4	As a user, I can register for the applicationthrough Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application byentering email & password		High	Sprint-1
	Dashboard	USN-5	As a user, I can access my dashboard aftersigning in.	I can access my account /dashboard	High	Sprint-1
Customer (Webuser)	Access	USN-6	As a user, I can setup a profile, and basicdetails by signing in.			
		USN-7	As a user, I will upload my resume, certificates, and other requirements.	I can perform several task inthe application	Medium	Sprint-1
Customer Care Executive	Chatbot	USN-8	As a user, I can seek guidance from the customer care executive.		High	Sprint-1
Administrator	DBMS	USN-9	As a administrator, I can keep the applicationsof your organization relies on running.	I can perform various modifications in the applications.	High	Sprint-1

Fig: 5.1.3

CHAPTER 6

PROJECT DESIGN

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for application by enteringmy email password and confirming it.	5	High	Obulipurusothaman k,Selvakumar R
Sprint-1		USN-2	As a user, I will receive confirmation email once Ihave registered for the application	5	High	Sowndhar S, Saravanan G
Sprint-1	Login	USN-3	As a user, I can log into the application by enteringemail & password	5	Medium	Saravanan G,Sowndhar S, Selvakumar R
Sprint-2	Post Job	USN-6	As a recruiter, I must post the job vacancy withdescription	6	High	Obulipurusothaman K, Sowndhar S,Saravanan G
Sprint-2	Job Search	USN-4	As a job seeker, I can search for the desiredcompanies	9	High	Sowndhar S,Selvakumar R
Sprint-3	Apply	USN-5	As a job seeker, I can apply for a company	6	High	Saravanan G, Obulipurusothaman K
Sprint-3	Send Confirmation	USN-7	Confirmation mail is sent from the respectedcompany	4	High	Obulipurusothaman K,Sowndhar S
Sprint-4	Dashboard	USN-8	As a user, I need to maintain the my actions in anapplication	6	High	Saravanan G, Selvakumar R
Sprint-4	Recruiter Review	USN -9	As a recruiter, I must make the reviews appear onthe candidate's profile	3	High	Obulipurusothaman K
Sprint-4	Chatbot	USN-10	As a user, I can interact with Watson Assistant toresolve my queries on skills to be learnt	1	Low	Selvakumar R ,Sowndhar S

Fig: 6.1.1

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	15	7 Days	24 Oct 2022	31 Oct 2022	15	31 Oct 2022
Sprint-2	15	7 Days	1 Nov 2022	07 Nov 2022	15	07 Nov 2022
Sprint-3	10	5 Days	08 Nov 2022	12 Nov 2022	10	12 Nov 2022
Sprint-4	10	5 Days	14 Nov 2022	19 Nov 2022	10	19 Nov 2022

Fig: 6.1.2

6.3 REPORTS FROM JIRA:

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

CHAPTER 7

CODING & SOLUTIONING
(Explain the features added in the project along with code)

7.1 Feature 1:

```python
import ibm_db
from flask import Flask, flash, redirect, render_template, request, url_for
from flask_cors import CORS, cross_origin
from flask import session
import sqlite3
import os

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;

con=sqlite3.connect("myimage.db")
con.execute("create table if not exists image(pid integer primary key,img TEXT)")
con.close()

app = Flask(__name__)
app.secret_key = '//sd_5#y2L"F4Q8z\n\xec]/'

app.config['UPLOAD_FOLDER']="static\images"

EMAIL=''

@app.route('/')
def index():
    return render_template('signup.html')

@app.route('/home')
def home():
    login=False
    if 'usernameid' and 'passwordid' in session:
        login=True
    return render_template('index.html', login=login)

@app.route('/signup', methods=['POST','GET'])
@cross_origin()
def signup():
```

```python
        EMAIL=email
        sql="SELECT * FROM authentication WHERE email=? AND password=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,password)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        if account:
            #session['loggedin'] = True
            #session['email'] = email
           #return render_template('home.html')
            return redirect(url_for('home'))
        else:
            error = "Invalid email/password"
            return redirect(url_for('login'))
    elif request.method=='GET':
        return render_template('login.html')

@app.route('/forgot', methods=['POST','GET'])
def forgot():
    if request.method=='POST':
        email=request.form['email']
        remail=email
        secret=request.form['secret']
        sql="SELECT * FROM authentication WHERE email=? AND secret=?"
        stmt=ibm_db.prepare(conn,sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.bind_param(stmt,2,secret)
        ibm_db.execute(stmt)
        account=ibm_db.fetch_assoc(stmt)
        if account:
            return redirect(url_for('reset'))
        else:
            return redirect(url_for('forgot'))
```

```python
def post():
    con = sqlite3.connect("myimage.db")
    con.row_factory = sqlite3.Row
    cur = con.cursor()
    cur.execute("select * from image")
    data = cur.fetchall()
    con.close()

    if request.method=='POST':
        upload_image=request.files['upload_image']

        if upload_image.filename!='':
            filepath=os.path.join(app.config['UPLOAD_FOLDER'],upload_image.filename)
            upload_image.save(filepath)
            con=sqlite3.connect("myimage.db")
            cur=con.cursor()
            cur.execute("insert into image(img)values(?)",(upload_image.filename,))
            con.commit()
            flash("File Upload Successfully","success")

            con = sqlite3.connect("myimage.db")
            con.row_factory=sqlite3.Row
            cur=con.cursor()
            cur.execute("select * from image")
            data=cur.fetchall()
            con.close()
            return render_template("post.html",data=data)
    return render_template("post.html",data=data)

@app.route('/delete_record/<string:id>')
def delete_record(id):
    try:
        con=sqlite3.connect("myimage.db")
        cur=con.cursor()
        cur.execute("delete from image where pid=?",[id])
```

```html
{% extends 'base.html' %}

{%  block head %}
<title>Home - Instant job</title>
<link rel="stylesheet" href="static/css/style.css">
<link rel="stylesheet" href="static/css/index.css">
{% endblock %}


{% block content %}



    <div class="topbar">

        <div class="topbarLeft">
            <h2 class="topbarHead">INSTANT JOB</h2>
        </div>

        <div class="topbarCenter">
            <ul class="topbarList">
                <li class="topbarListItem"><a class="topbarListItem" href="contacts">Contacts</a></li>
                <li class="topbarListItem"><a class="topbarListItem" href="login">Log in</a></li>
                <li class="topbarListItem"><a class="topbarListItem" href="signup">Sign up</a></li>

            </ul>
        </div>

    </div>

    <div class="header">

        <div class="headerImg">
            <h3 class="headerTitle">WELCOME TO INSTANT JOB</h3>

```

```python
import ibm_db
from flask import Flask, flash, redirect, render_template, request, url_for
from flask_cors import CORS, cross_origin
from flask import session
import sqlite3
import os

conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;

con=sqlite3.connect("myimage.db")
con.execute("create table if not exists image(pid integer primary key,img TEXT)")
con.close()

app = Flask(__name__)
app.secret_key = '//sd_5#y2L"F4Q8z\n\xec]/'

app.config['UPLOAD_FOLDER']="static\images"

EMAIL=''

@app.route('/')
def index():
    return render_template('signup.html')

@app.route('/home')
def home():
    login=False
    if 'usernameid' and 'passwordid' in session:
        login=True
    return render_template('index.html', login=login)

@app.route('/signup', methods=['POST','GET'])
@cross_origin()
def signup():
```

Fig: 7.1.1

7.2 Feature 2:

```html
<input class="shadow form-control" type="text" id="curcityid" name="curcity" required><br><br>
<label for="curstateid">State:</label>
<input class="shadow form-control" type="text" id="curstateid" name="curstate" required><br><br>
<label for="curcntryid">Country:</label>
<input class="shadow form-control" type="text" id="curcntryid" name="curcntryid" required><br><br>
</fieldset><br>

<fieldset>
<legend>QUALIFICATIONS</legend><br>
<table>
<tr>
<td align="center"><b>Sl.No.</b></td>
<td align="center"><b>Degree</b></td>
<td align="center"><b>Board</b></td>
<td align="center"><b>Percentage</b></td>
<td align="center"><b>Year of Passing</b></td>
</tr>

<tr>
<td>1</td>
<td>Class X</td>
<td><input type="text" name="Xboard" required></td>
<td><input type="text" name="XPercent" required></td>
<td><input type="number" name="XYOP" required></td>
</tr>

<tr>
<td>2</td>
<td>Class XII</td>
<td><input type="text" name="XIIboard" required></td>
<td><input type="text" name="XIIPercent" required></td>
<td><input type="number" name="XIIYOP" required></td>
</tr>

<tr>
```

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

<head>
    <meta charset="utf-8">
    <title>Instant job</title>
    <meta content="width=device-width, initial-scale=1.0" name="viewport">
    <meta content="" name="keywords">
    <meta content="" name="description">

        <!-- Favicon -->
        <link href="img/favicon.ico" rel="icon">

        <!-- Google Web Fonts -->
        <link rel="preconnect" href="https://fonts.googleapis.com">
        <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
        <link href="https://fonts.googleapis.com/css2?family=Heebo:wght@400;500;600&family=Inter:wght@700;800&display=swap" rel="stylesheet"
        >

        <!-- Icon Font Stylesheet -->
        <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
        <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">

        <!-- Libraries Stylesheet -->
        <link href="lib/animate/animate.min.css" rel="stylesheet">
        <link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">

        <!-- Customized Bootstrap Stylesheet -->
        <link href="css/bootstrap.min.css" rel="stylesheet">



    <link rel="stylesheet" href="static/css/style.css">
    <link rel="stylesheet" href="static/css/index.css">
</head>
```

app.py — D:\IBM PROJECT\sprint1 | app.py — C:\...\ibm project - final | forgot.html | jobApplication.html | jobList.html | reset.html

```html
{% extends 'base.html' %}

{%  block head %}
    <title>Forgot - Instant job</title>
    <link rel="stylesheet" href="static/css/style.css">
    <link rel="stylesheet" href="static/css/index.css">
    {% endblock %}


{% block content %}

    <div class="topbar">

        <div class="topbarLeft">
            <h2 class="topbarHead">INSTANT JOB</h2>
        </div>

        <div class="topbarCenter">
            <ul class="topbarList">
                <li class="topbarListItem"><a class="topbarListItem" href="signup">Sign up</a></li>
                <li class="topbarListItem"><a class="topbarListItem" href="login">Log in</a></li>

            </ul>
        </div>
    </div>

    <section class="py-4 py-md-5 mt-5">
        <div class="container py-md-5">
            <div class="row d-flex align-items-center">
                <div class="col-md-5 col-xl-4 text-center text-md-start">
                    <h2 class="display-6 mb-4">Reset your password</h2>
                    <p class="text-muted">Enter email and new password.</p>
                    <form method="post">
                        <div class="mb-3"><input class="shadow form-control" type="email" name="email" placeholder="Email"></div>
                        <div class="mb-3"><input class="shadow form-control" type="text" name="password" placeholder="Password"></div>
```

Fig: 7.2.1

CHAPTER 8

TESTING

8.1 TEST CASE:

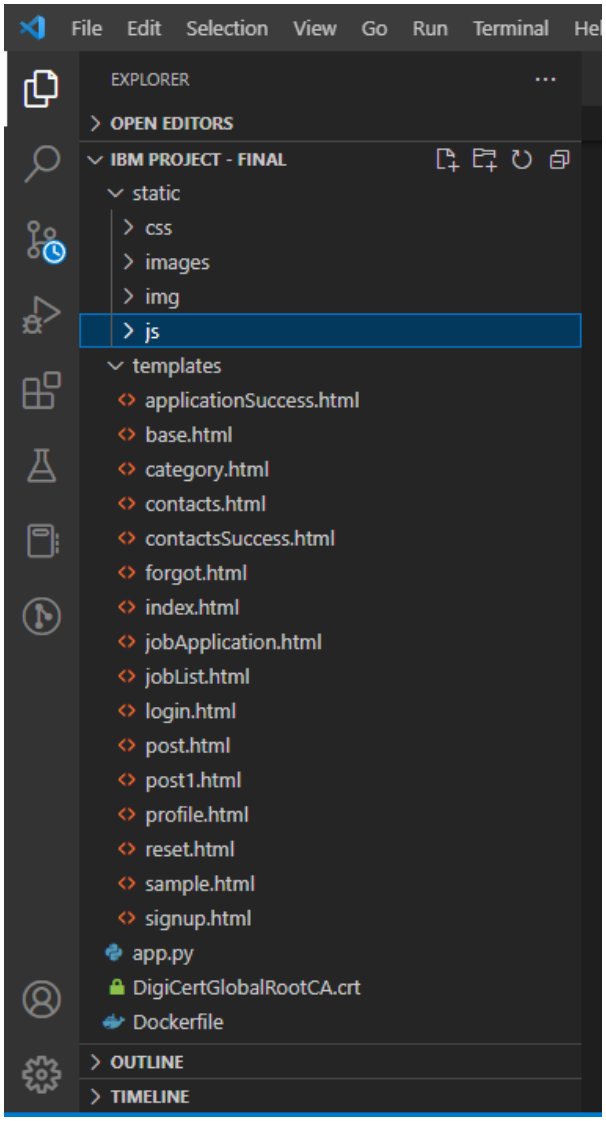


Fig: 8.1.1

8.2 USER ACCEPTANCE TESTING:

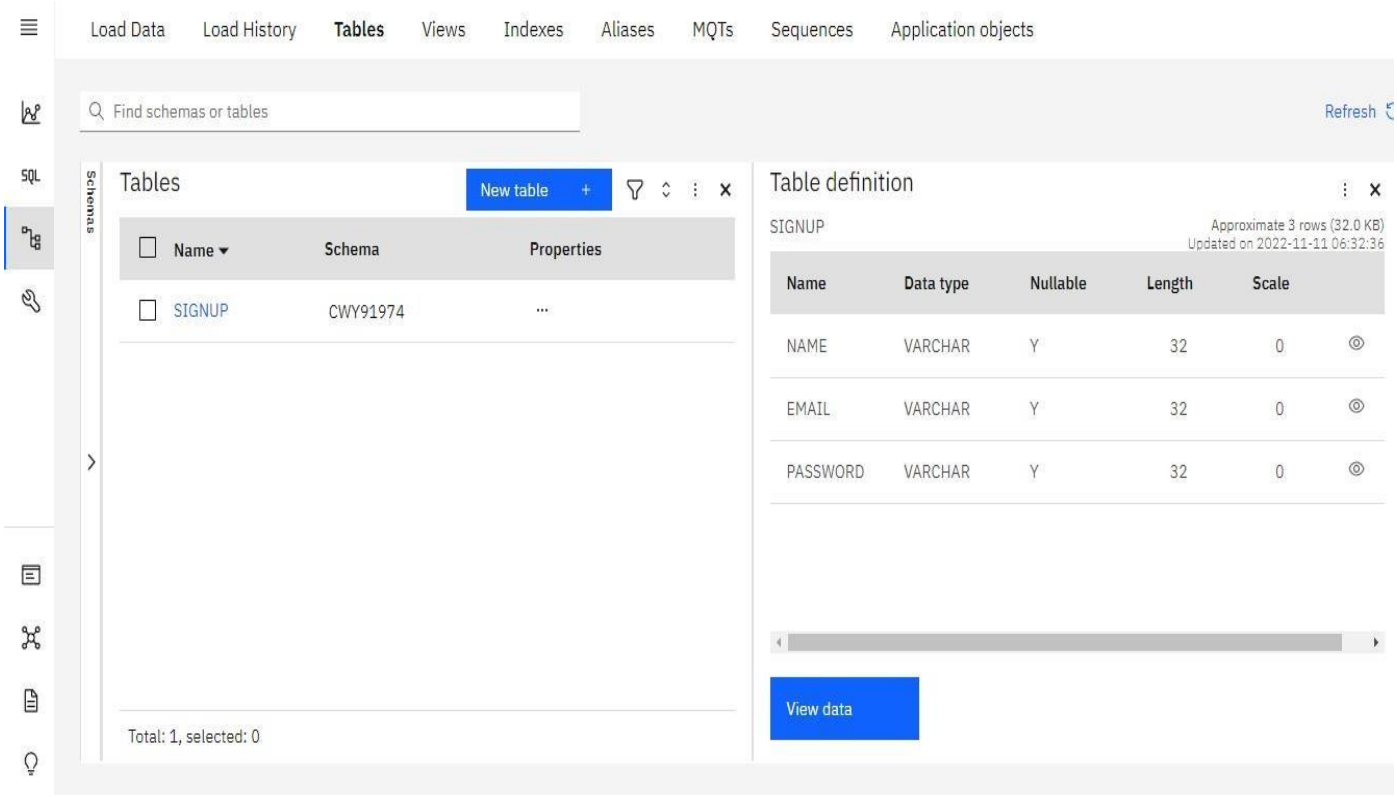


Fig: 8.2.1

CHAPTER 9

RESULTS

9.1 PERFORMANCE METRICS:

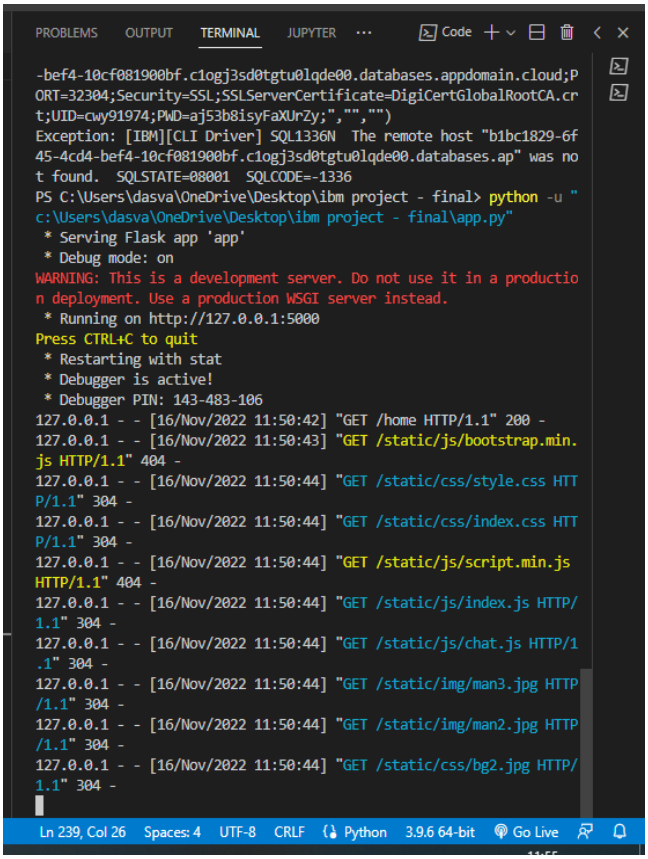
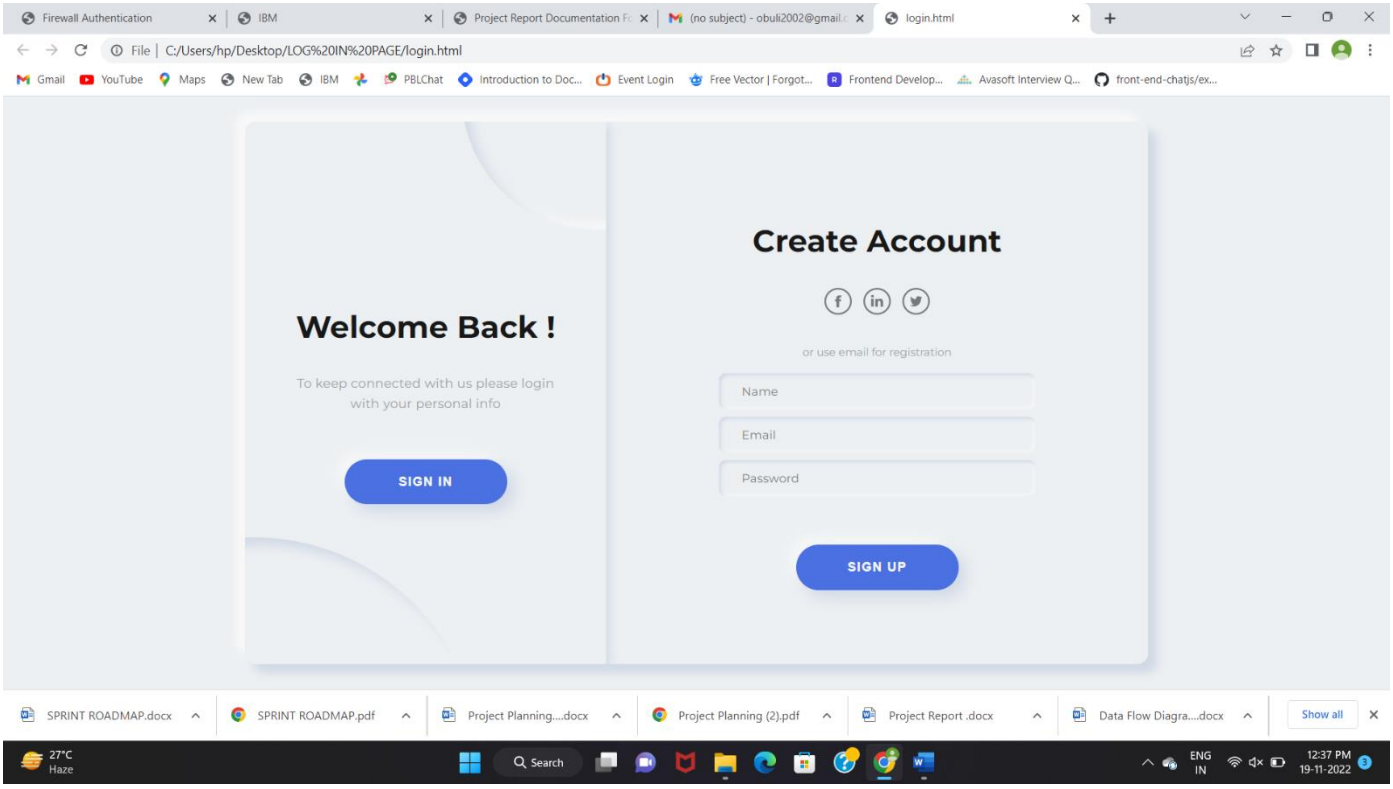
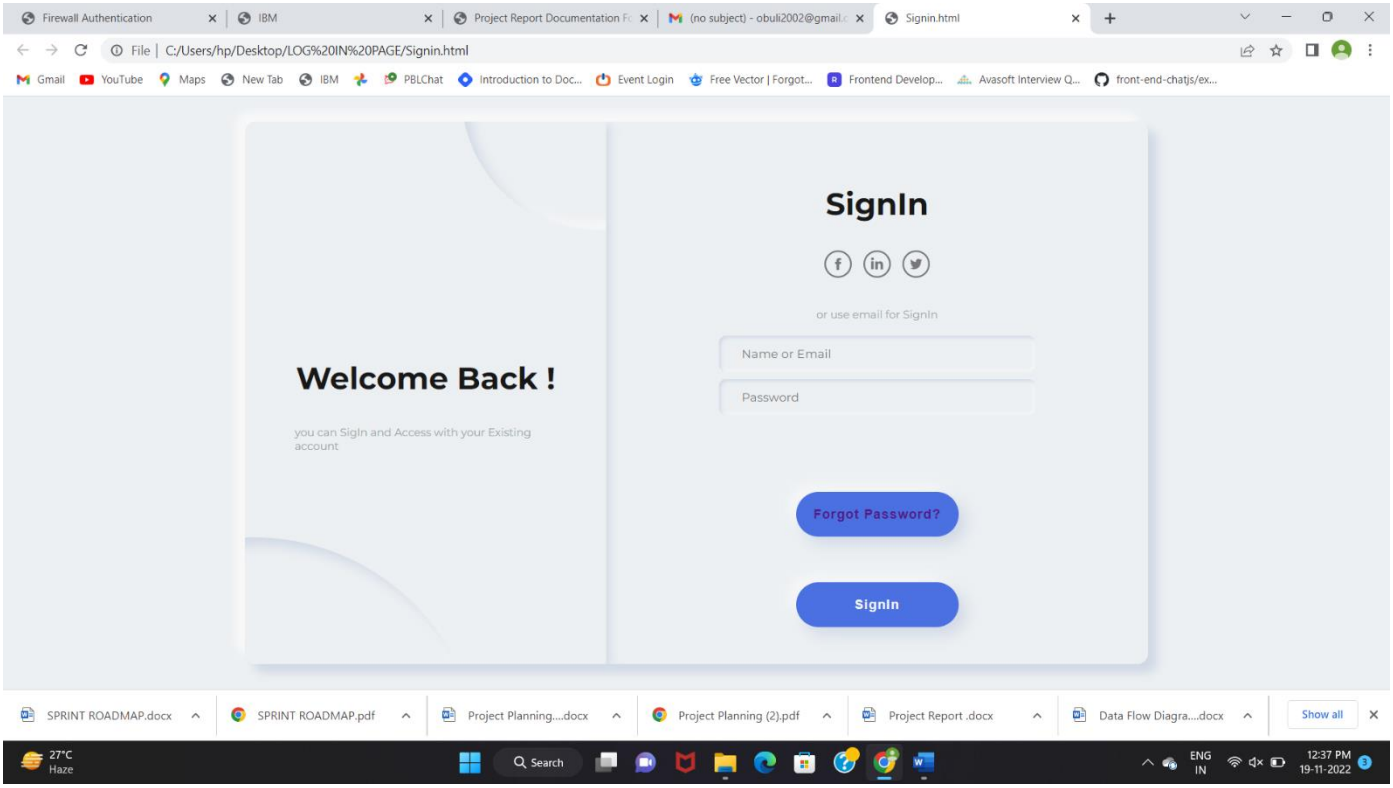
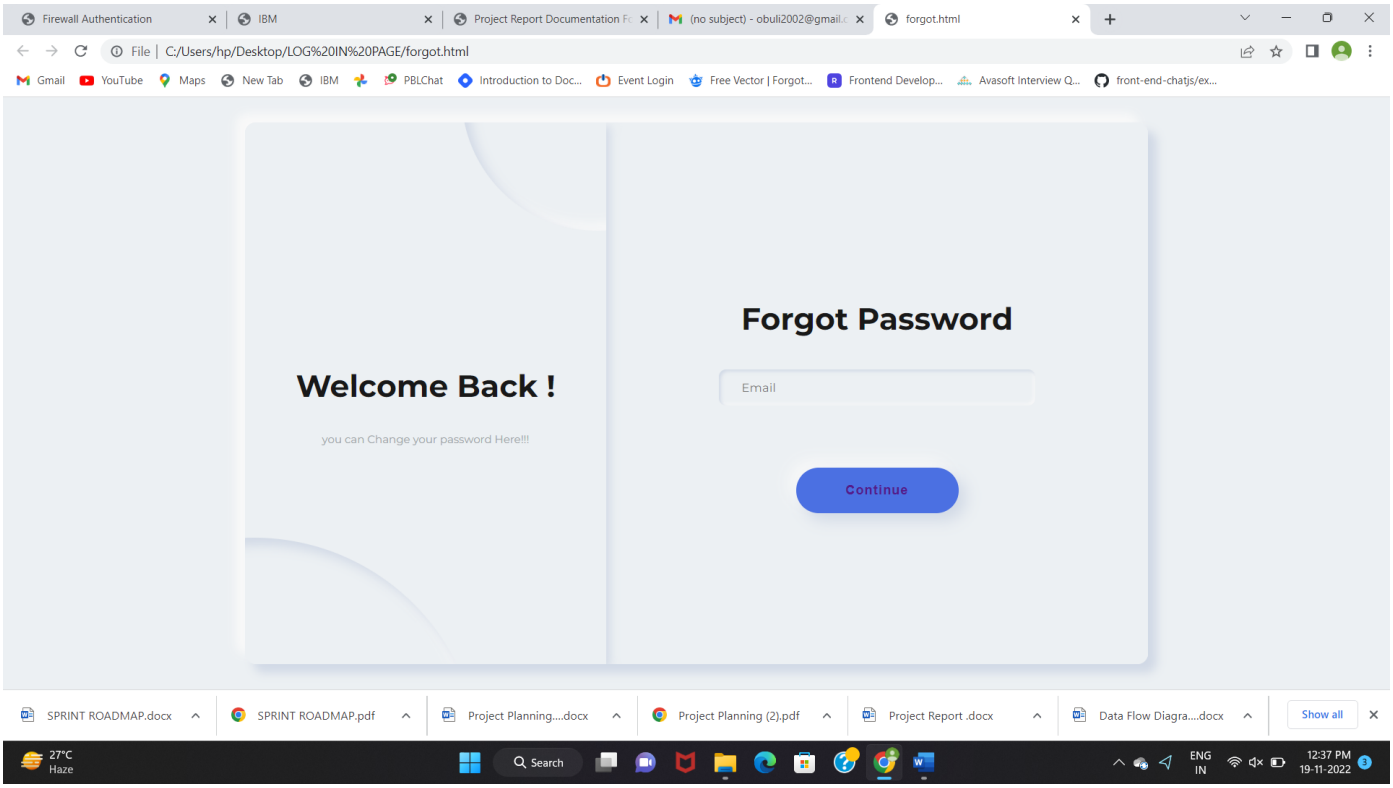


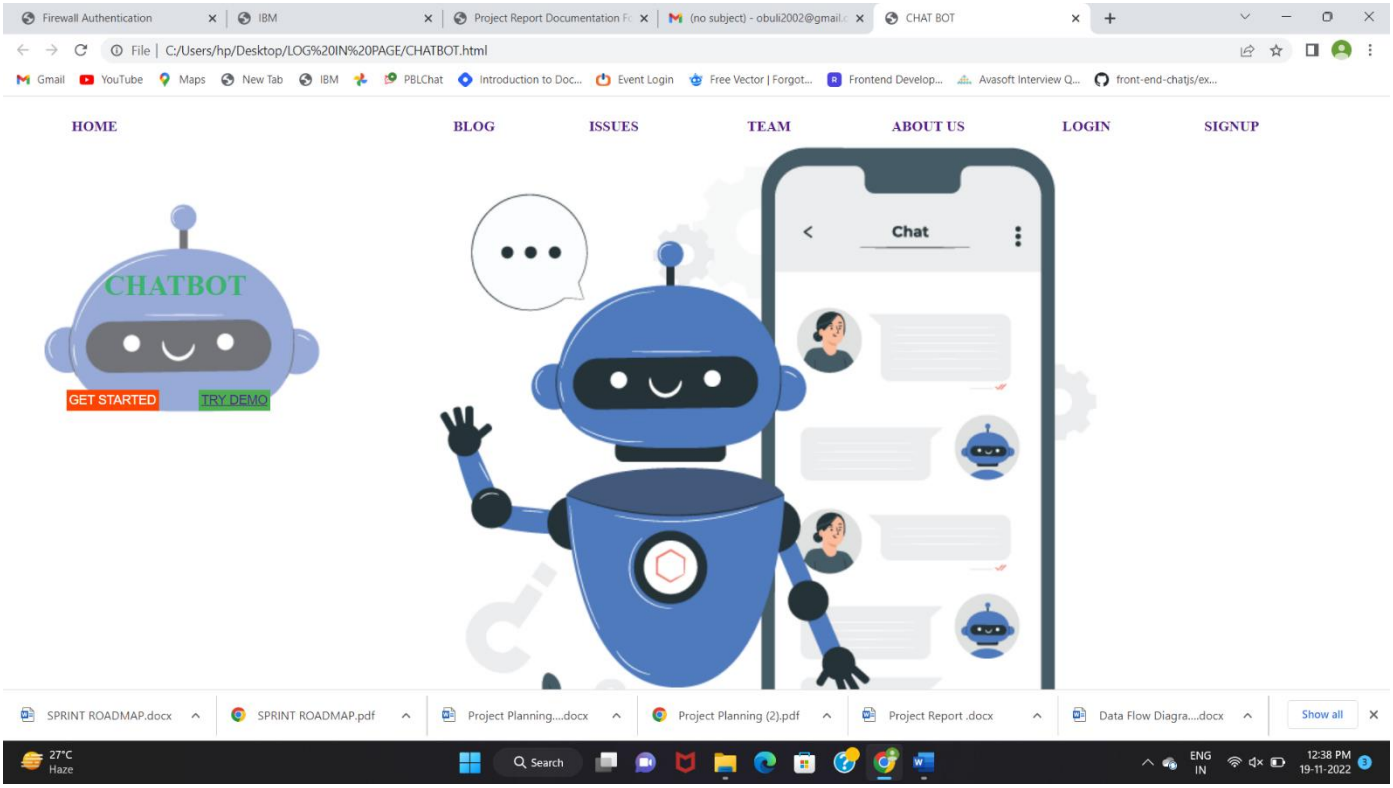
Fig: 9.1.1

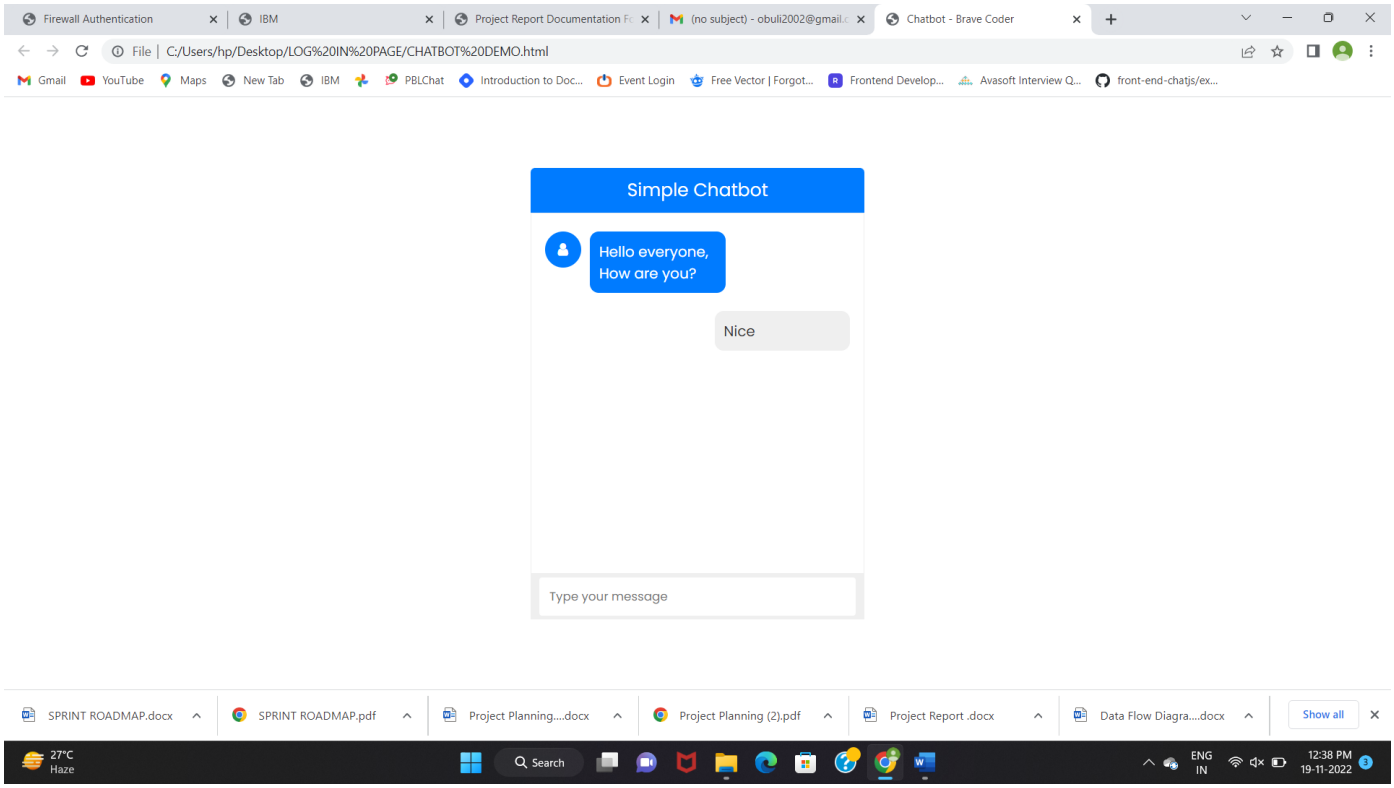
9.2 OUTPUT:

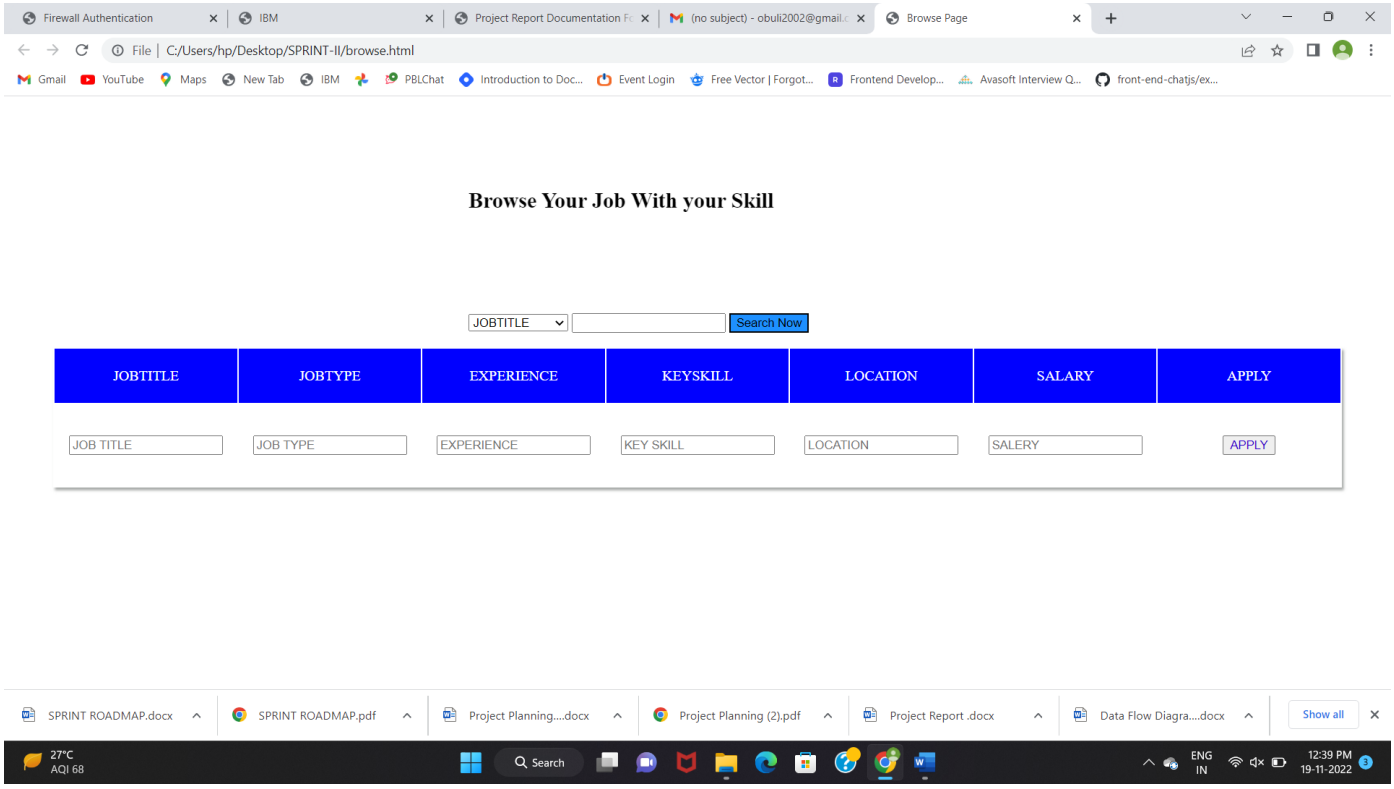

Firewall Authentication
IBM
Project Report Documentation F
(no subject) - obuli2002@gmail.c
login.html
File | C:/Users/hp/Desktop/LOG%20IN%20PAGE/login.html
Welcome Back !
To keep connected with us please login with your personal info
SIGN IN
Create Account
or use email for registration
Name
Email
Password
SIGN UP

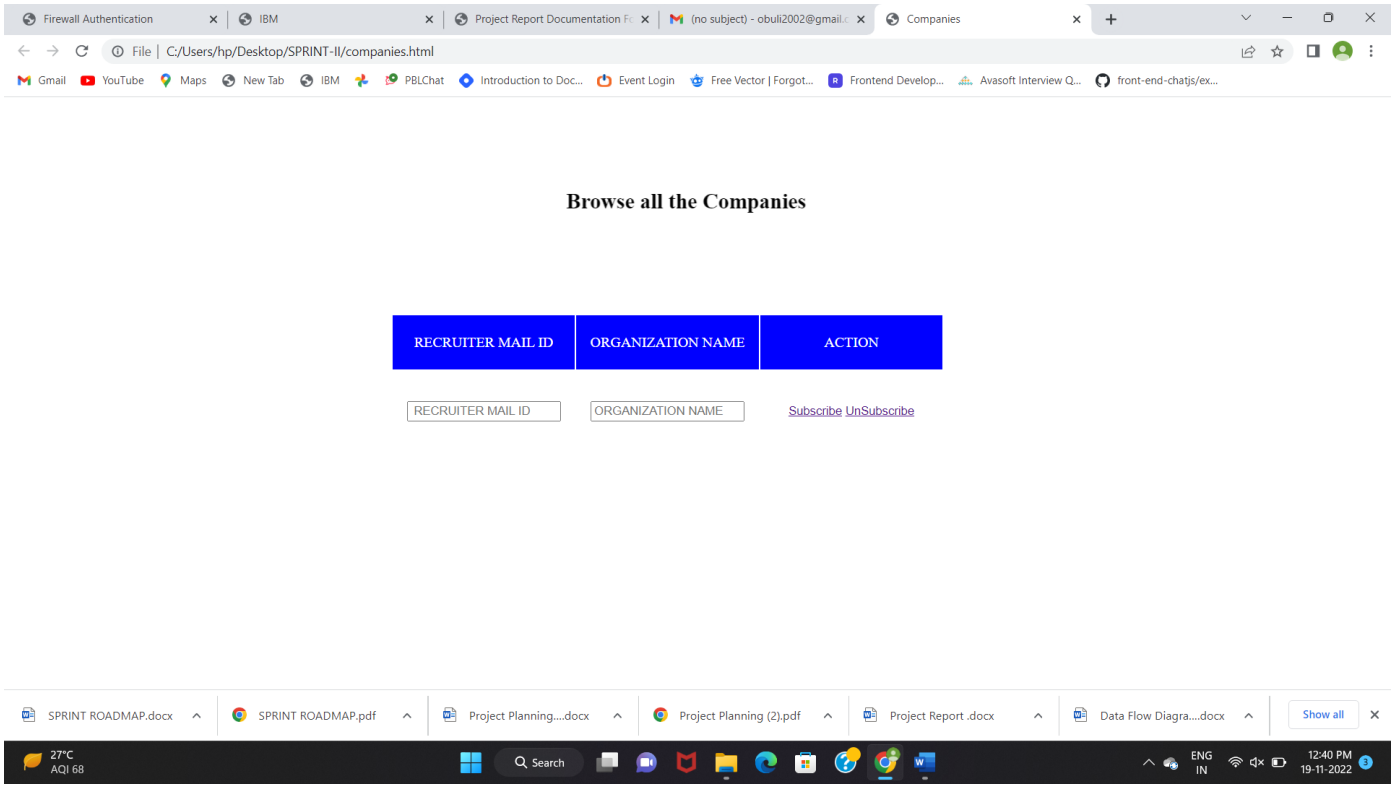

Firewall Authentication
IBM
Project Report Documentation F
(no subject) - obuli2002@gmail.c
Signin.html
File | C:/Users/hp/Desktop/LOG%20IN%20PAGE/Signin.html
Welcome Back !
you can SigIn and Access with your Existing account
SignIn
or use email for SignIn
Name or Email
Password
Forgot Password?
SignIn

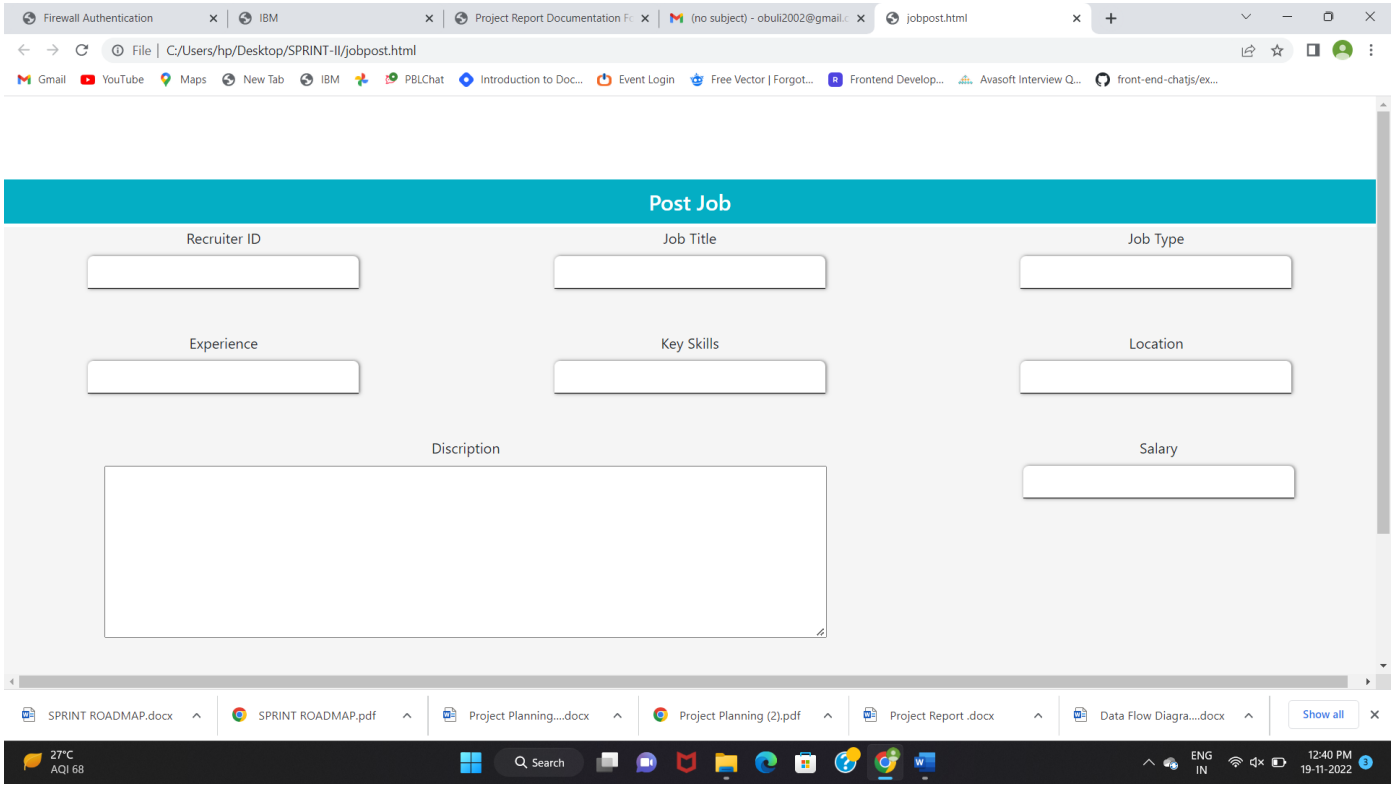

Firewall Authentication
IBM
Project Report Documentation Fo
(no subject) - obuli2002@gmail.c
forgot.html
File | C:/Users/hp/Desktop/LOG%20IN%20PAGE/forgot.html
Welcome Back !
you can Change your password Here!!!
Forgot Password
Email
Continue
SPRINT ROADMAP.docx
SPRINT ROADMAP.pdf
Project Planning....docx
Project Planning (2).pdf
Project Report .docx
Data Flow Diagra....docx
Show all


Firewall Authentication
IBM
Project Report Documentation Fo
(no subject) - obuli2002@gmail.c
CHAT BOT
File | C:/Users/hp/Desktop/LOG%20IN%20PAGE/CHATBOT.html
HOME
BLOG
ISSUES
TEAM
ABOUT US
LOGIN
SIGNUP
CHATBOT
GET STARTED
TRY DEMO
Chat
SPRINT ROADMAP.docx
SPRINT ROADMAP.pdf
Project Planning....docx
Project Planning (2).pdf
Project Report .docx
Data Flow Diagra....docx
Show all


Firewall Authentication
IBM
Project Report Documentation Fo
(no subject) - obuli2002@gmail.c
Chatbot - Brave Coder
File | C:/Users/hp/Desktop/LOG%20IN%20PAGE/CHATBOT%20DEMO.html
Simple Chatbot
Hello everyone, How are you?
Nice
Type your message
SPRINT ROADMAP.docx
SPRINT ROADMAP.pdf
Project Planning....docx
Project Planning (2).pdf
Project Report .docx
Data Flow Diagra....docx
Show all


Firewall Authentication
IBM
Project Report Documentation Fo
(no subject) - obuli2002@gmail.c
Browse Page
File | C:/Users/hp/Desktop/SPRINT-II/browse.html
Browse Your Job With your Skill
JOBTITLE
Search Now
JOBTITLE
JOBTYPE
EXPERIENCE
KEYSKILL
LOCATION
SALARY
APPLY
JOB TITLE
JOB TYPE
EXPERIENCE
KEY SKILL
LOCATION
SALERY
APPLY
SPRINT ROADMAP.docx
SPRINT ROADMAP.pdf
Project Planning....docx
Project Planning (2).pdf
Project Report .docx
Data Flow Diagra....docx
Show all


Firewall Authentication
IBM
Project Report Documentation Fo
(no subject) - obuli2002@gmail.c
Companies
File | C:/Users/hp/Desktop/SPRINT-II/companies.html
Gmail YouTube Maps New Tab IBM PBLChat Introduction to Doc... Event Login Free Vector | Forgot... Frontend Develop... Avasoft Interview Q... front-end-chatjs/ex...
Browse all the Companies
RECRUITER MAIL ID
ORGANIZATION NAME
ACTION
RECRUITER MAIL ID
ORGANIZATION NAME
Subscribe UnSubscribe
SPRINT ROADMAP.docx
SPRINT ROADMAP.pdf
Project Planning....docx
Project Planning (2).pdf
Project Report .docx
Data Flow Diagra....docx
Show all
27°C AQI 68
Search
ENG IN
12:40 PM 19-11-2022


Firewall Authentication
IBM
Project Report Documentation Fo
(no subject) - obuli2002@gmail.c
jobpost.html
File | C:/Users/hp/Desktop/SPRINT-II/jobpost.html
Gmail YouTube Maps New Tab IBM PBLChat Introduction to Doc... Event Login Free Vector | Forgot... Frontend Develop... Avasoft Interview Q... front-end-chatjs/ex...
Post Job
Recruiter ID
Job Title
Job Type
Experience
Key Skills
Location
Discription
Salary
SPRINT ROADMAP.docx
SPRINT ROADMAP.pdf
Project Planning....docx
Project Planning (2).pdf
Project Report .docx
Data Flow Diagra....docx
Show all
27°C AQI 68
Search
ENG IN
12:40 PM 19-11-2022

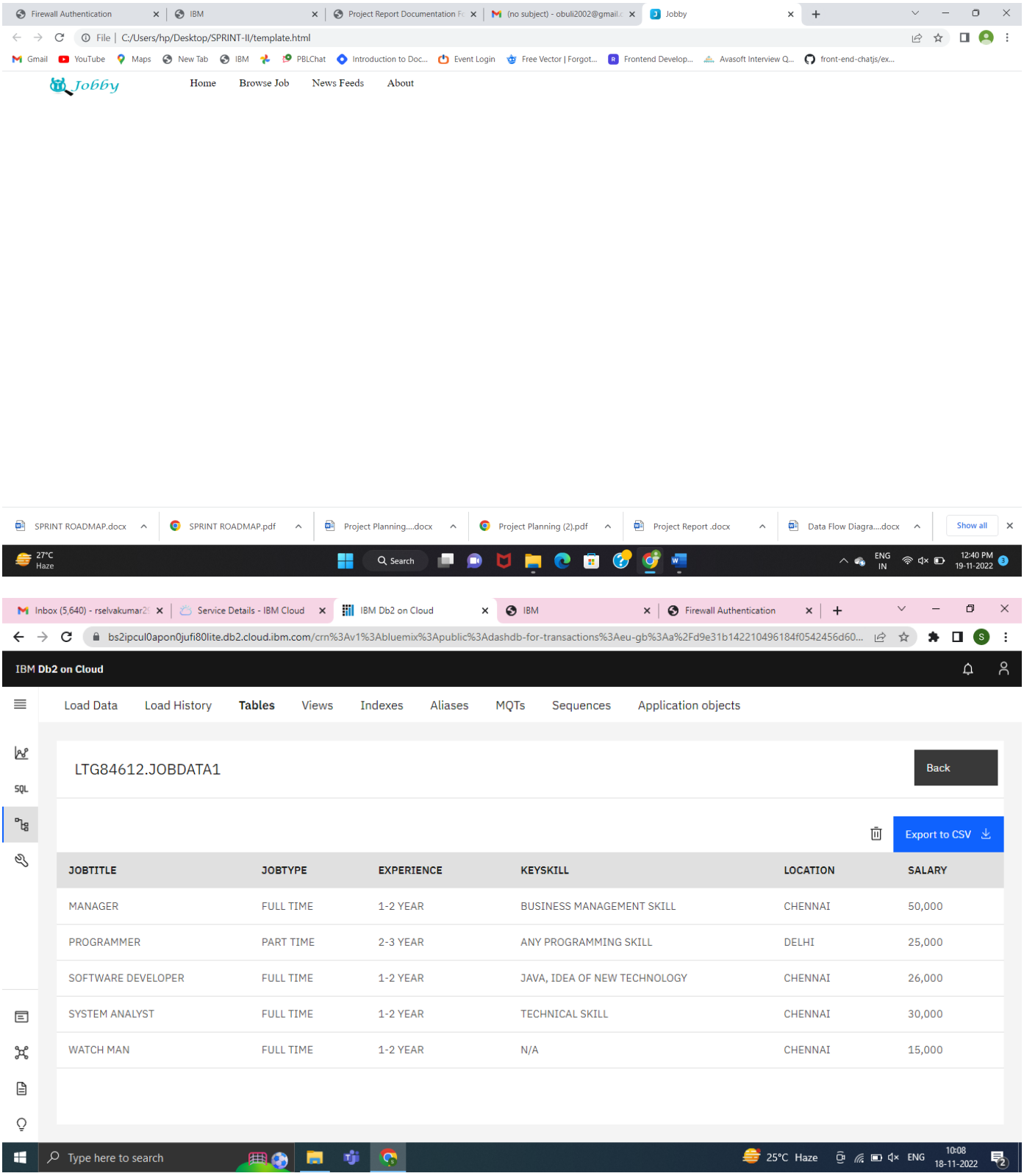

Jobby
Home
Browse Job
News Feeds
About
IBM Db2 on Cloud
Load Data
Load History
Tables
Views
Indexes
Aliases
MQTs
Sequences
Application objects
LTG84612.JOBDATA1
Back
Export to CSV
JOBTITLE JOBTYPE EXPERIENCE KEYSKILL LOCATION SALARY
MANAGER FULL TIME 1-2 YEAR BUSINESS MANAGEMENT SKILL CHENNAI 50,000
PROGRAMMER PART TIME 2-3 YEAR ANY PROGRAMMING SKILL DELHI 25,000
SOFTWARE DEVELOPER FULL TIME 1-2 YEAR JAVA, IDEA OF NEW TECHNOLOGY CHENNAI 26,000
SYSTEM ANALYST FULL TIME 1-2 YEAR TECHNICAL SKILL CHENNAI 30,000
WATCH MAN FULL TIME 1-2 YEAR N/A CHENNAI 15,000

Fig: 9.2.1

CHAPTER 10

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

- When recruiting externally, hiring teams find candidates, evaluate them and, if all goes well, persuade them to join their company. All of which takes time.
- Everyone needs some time to adjust to a new role, but internal hires are quicker to onboard than external hires.
- May be familiar with people in their new team, especially in smaller businesses.
- Know how your company operates and most of your policies and practices.

DISADVANTAGES

- Employees who were considered for a role could feel resentful if a colleague or external candidate is eventually hired.
- While your company may have a lot of qualified candidates for specific positions, this isn't necessarily true for every open

CHAPTER 11

CONCLUSION

CONCLUSION

By the end of this project we will

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

CHAPTER 12

FUTURE SCOPE

FUTURE SCOPE

- AI is revolutionizing the recommender systems.
- The popularity of LinkedIn and Google for jobs has proved that there is a future for job boards if effectively managed to provide solutions
- Right pricing strategies for online recruitment advertising are essential to get an effective response.

Recruiters and job seekers are experiencing an entirely automated process of searching and connecting. All job boards should be perfectly indexed, highly responsive, and exhaustive in job descriptions to establish their credibility and reliability

CHAPTER 13

APPEINDEX

REF: https://github.com/IBM-EPBL/IBM-Project-32088-1660207990