IBM – NALAIYA THIRAN PROJECT

SKILL JOB RECOMMENDER APPLICATION

INDUSTRY MENTOR : KRISHNA CHAITANYA

FACULTY MENTOR : ROHINI M

TEAM ID : PNT2022TMID02696

TEAM LEAD : DHIKSHA S

TEAM MEMBER : ELAMMATHI M

TEAM MEMBER : ADAIKALA

PITCHAI THABITHA J I

TEAM MEMBER : EMMA THOMAS

ABSTRACT

In the last years, job recommender systems have become popular since they successfully reduce information overload by generating personalized job suggestions. Although in the literature exists a variety of techniques and strategies used as part of job recommender systems, most of them fail to recommending job vacancies that fit properly to the job seekers profiles. Thus, the contributions of this work are threefold, we: i) made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites; ii) put forward the proposal of a framework for job recommendation based on professional skills of job seekers; and iii) carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. We thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue.

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

TABLE OF CONTENT

CHAPTER	CONTENTS	PAGE NO
1	**INTRODUCTION** 1.1 PROJECT OVERVIEW 1.2 PURPOSE	**05**
2	**LITERATURE SURVEY** 2.1 EXISTING PROBLEM 2.2 REFERENCES 2.3 PROBLEM STATEMENT DEFINITION	**06**
3	**IDEATION & PROPOSED SOLUTION** 3.1 EMPATHY MAP CANVAS 3.2 IDEATION & BRAINSTROMING 3.3 PROPOSED SOLUTION 3.4 PROBLEM SOLUTION FIT	**09**
4	**REQUIREMENT ANALYSIS** 4.1 FUNCTIONAL REQUIREMENT 4.2 NON-FUNCTIONAL REQUIREMENTS	**15**
5	**PROJECT DESIGN** 5.1 DATA FLOW DIAGRAMS 5.2 SOLUTION & TECHNICAL ARCHITECTURE 5.3 USER STORIES	**17**
6	**PROJECT PLANNING & SCHEDULING** 6.1 SPRINT PLANNING & ESTIMATION 6.2 SPRINT DELIVERY SCHEDULE 6.3 REPORTS FROM JIRA	**20**

7	**CODING & SOLUTIONING** 7.1 FEATURE 1 7.2 FEATURE 2 7.3 DATABASE SCHEMA	**24**
8	**TESTING** 8.1 TEST CASES 8.2 USER ACCEPTANCE TESTING	**36**
9	**RESULTS** 9.1 PERFORMANCE METRICS	**38**
10	**ADVANTAGES & DISADVANTAGES**	**40**
11	**CONCLUSION**	**41**
12	**FUTURE SCOPE**	**42**

1. INTRODUCTION

1.1 PROJECT OVERVIEW

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

1.2 PURPOSE

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

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM:

Existing system is not very efficient, it does not benefit the user in maximum way, so the proposed system uses IBM cloud services like db2, Watson virtual assistant , cluster , Kubernetes and docker for containerization of the application.

They tend to miss out on these postings because there is an ocean of existing systems that list millions of jobs which are generally not relevant at all to the users. There is an abundance of choices and not much streamlining. On the basis of the actual skills or interests of an individual, job seekers often find themselves unable to find the appropriate employment for themselves. This system, therefore, approaches the idea from a data point of view, emphasizing more on the quality of the data than the quantity.

2.2 REFERENCES:

1. Shaha T Al-Otaibi and Mourad Ykhlef. “A survey of job recommender systems”. In: International Journal of the Physical Sciences 7.29 (2012), pp. 5127–5142. issn: 19921950. doi: 10.5897/IJPS12. 482
2. N Deniz, A Noyan, and O G Ertosun. “Linking Person-job Fit to Job Stress: The Mediating Effect of Perceived Person-organization Fit”. In: Procedia - Social and Behavioral Sciences 207 (2015), pp. 369– 376.
3. M Diaby, E Viennet, and T Launay. “Toward the next generation of recruitment tools: An online social network-based job recommender system”. In: Proc. of the 2013 IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining, ASONAM 2013 (2013), pp. 821–828. doi: 10. 1145/2492517.2500266.
4. M Diaby and E Viennet. “Taxonomy-based job recommender systems on Facebook and LinkedIn profiles”. In: Proc. of Int. Conf. on Research Challenges in Information Science (2014), pp. 1–6. issn: 21511357. doi: 10.1109/RCIS.2014.6861048.
5. M Kusner et al. “From word embeddings to document distances”. In: Proc. of the 32nd Int. Conf. on Machine Learning, ICML’15. 2015, pp. 957–966.
6. T Mikolov et al. “Distributed Representations of Words and Phrases and Their Compositionality”. In: Proc. of the 26th Int. Conf. on Neural Information Processing Systems - Volume 2. NIPS’13. Lake Tahoe, Nevada, 2013, pp. 3111–3119. url: http://dl.acm.org/citation.cfm?id=2999792. 2999959.
7. T Mikolov et al. “Efficient estimation of word representations in vector space”. In: arXiv preprint arXiv:1301.3781 (2013).
8. G Salton and C Buckley. “Term-weighting approaches in automatic text retrieval”. In: Information Processing and Management 24.5 (1988), pp. 513–523. issn: 0306-4573. doi: https://doi.org/10. 1016/0306- 4573(88)90021- 0. url: http://www.sciencedirect.com/science/article/pii/ 030645738890021

2.3 PROBLEM STATEMENT DEFINITION

"Can an efficient recommender system be modeled for the Job seekers which recommend Jobs with the user’s skill set and job domain and also addresses the issue of cold start?".

In current situation recruitment s done manually for lakhs of students in which many talented students may lose their opportunities due to different reasons since it is done manually, and company also need the highly talented people from the mass group for their growth. So we have build a cloud application to do this process in a efficient manner.

3. IDEATION & PROPOSED SOLUTION

In this project you will be working on two modules :

1. Admin and
2. User

ADMIN:

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

USER :

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

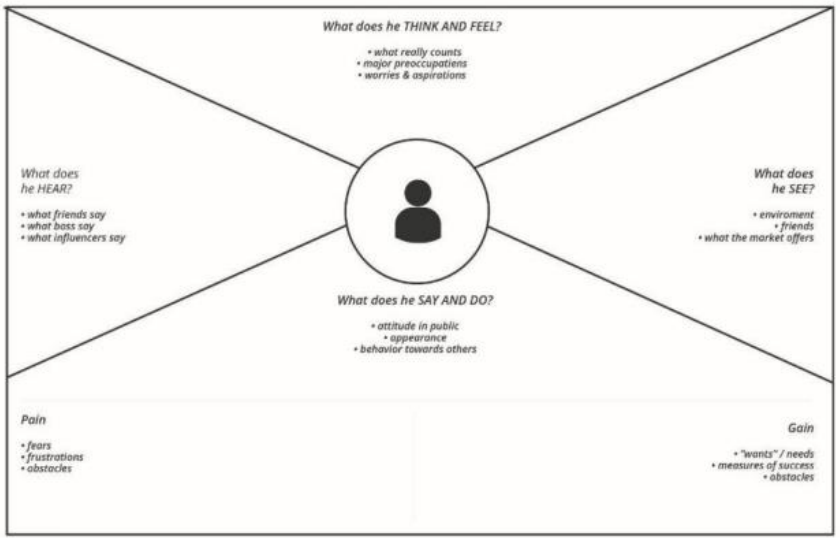
FEATURES OF CHATBOT :

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

3.1 EMPATHY MAP CANVAS:

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

1) create a shared understanding of user needs, and
2) aid in decision making



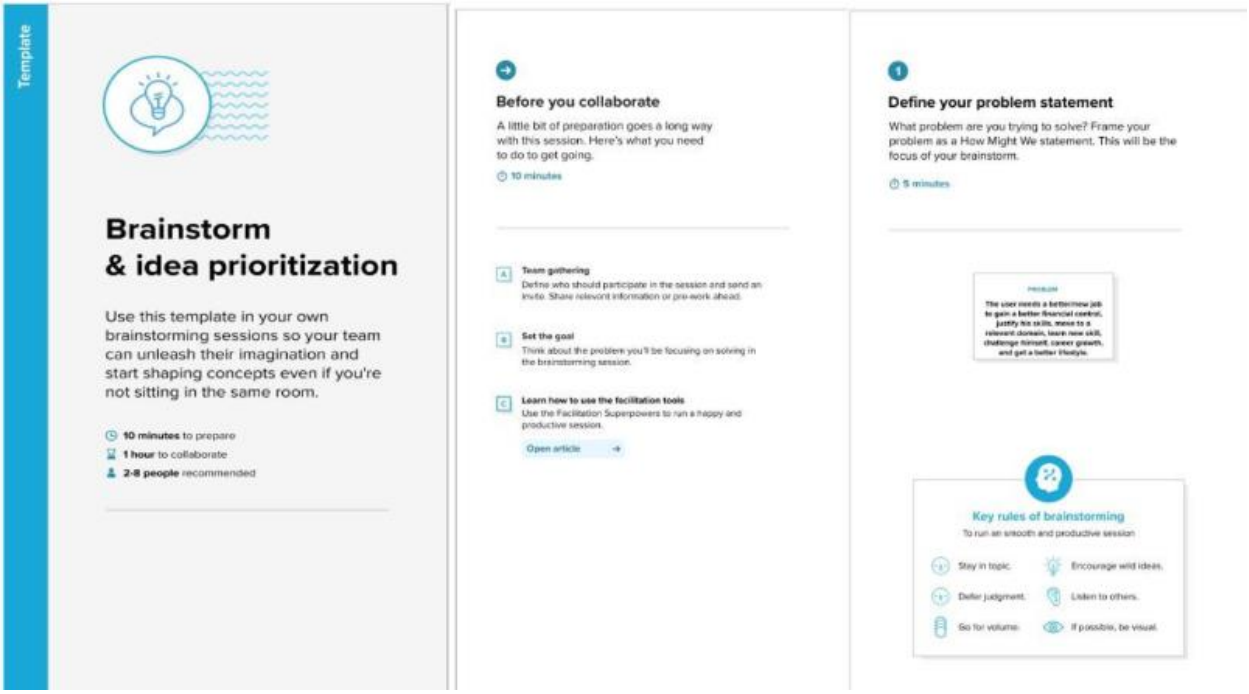
3.2 IDEATION & BRAINSTROMING:

Brainstorm & Idea Prioritization Template:

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

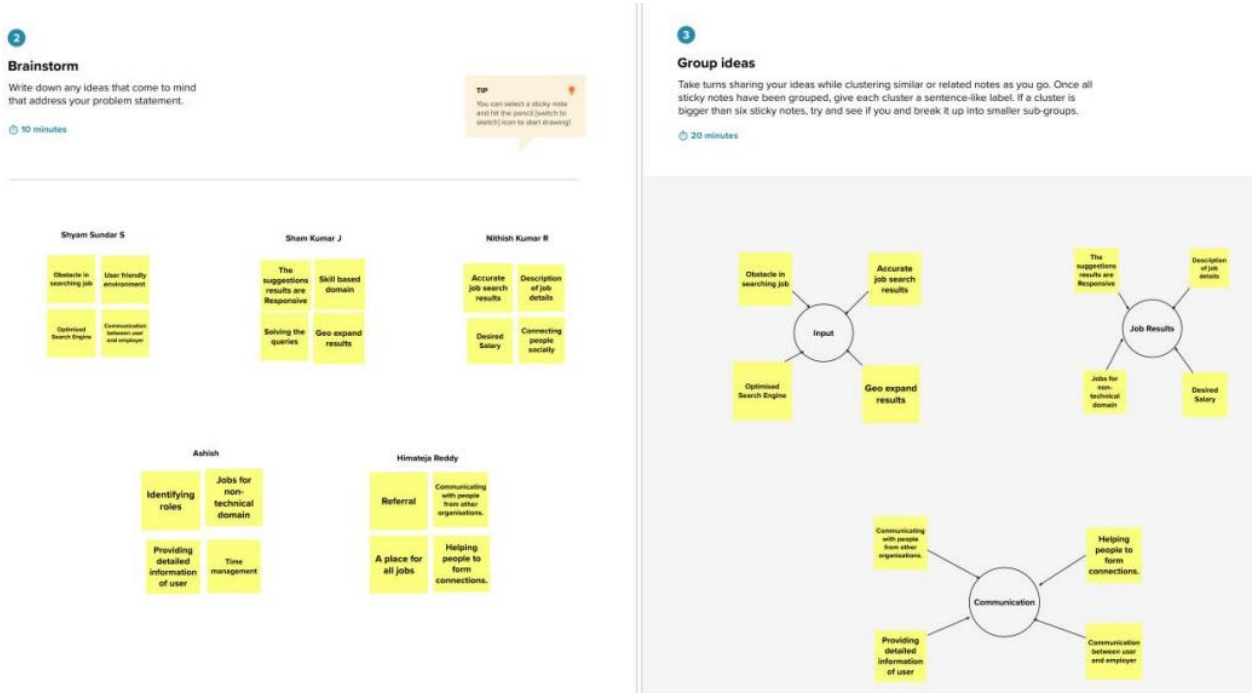
STEP 1:

Team Gathering, Collaboration and Select the Problem Statement



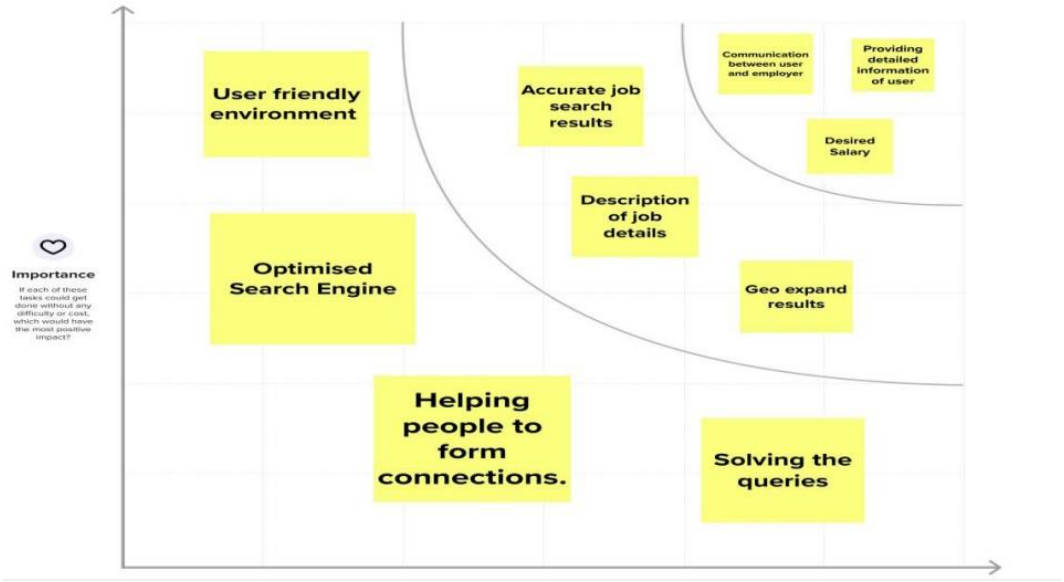
STEP 2:

Brainstorm, Idea Listing and Grouping



STEP 3:

Idea Prioritization



3.3 PROPOSED SOLUTION:

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

To develop an end-to-end web application capable of displaying the current job openings based on the user skillset. The user and their information are stored in the Database. An alert is sent when there is an opening based on the user skillset. Users will interact with the chatbot and can get the recommendations based on their skills. We can use a job search API to get the current job openings in the market which will fetch the data directly from the webpage

3.4 PROBLEM SOLUTION FIT

4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT:

Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
User Registration	Registration through Form Registration through Gmail
User Confirmation	Confirmation via Email Confirmation via OTP
Chat Bot	A Chat Bot will be there in website to solve user queries and problems related to applying a job, search for a job and much more.
User Login	Login through Form Login through Gmail
User Search	Exploration of Jobs based on job filters and skill recommendations.
User Profile	Updation of the user profile through the login credentials
User Acceptance	Confirmation of the Job.

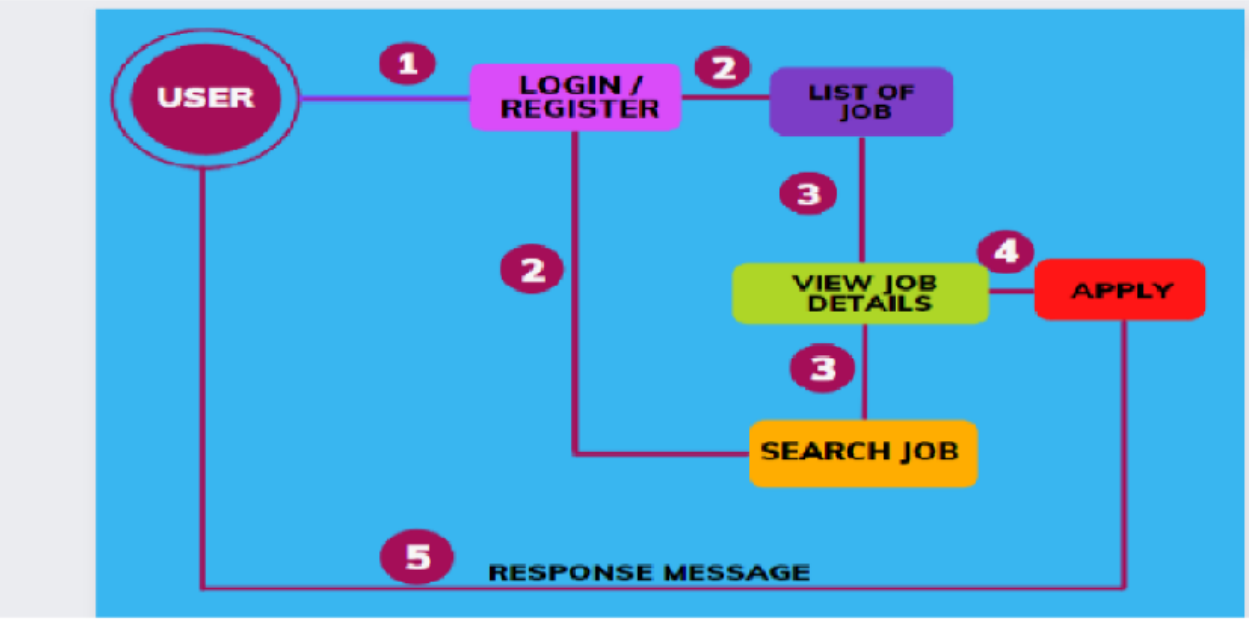
4.2 NON-FUNCTIONAL REQUIREMENTS:

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

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS:

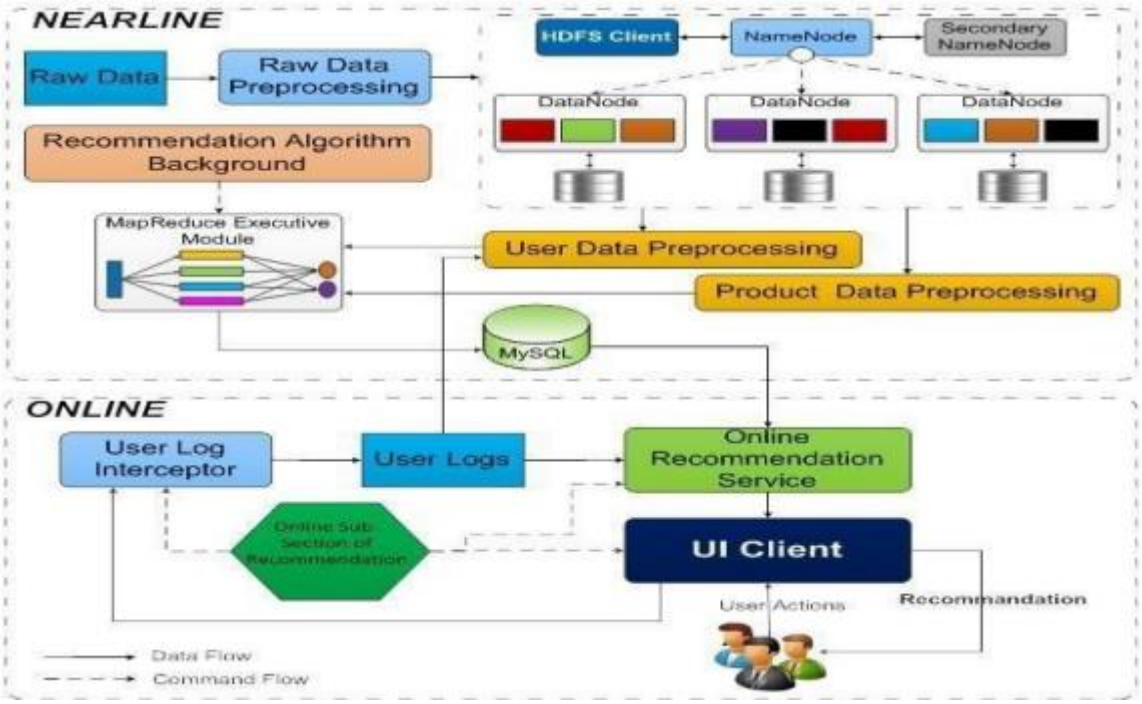
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE:

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

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed and delivered.
- Provide the best business require recommend by using the optimised and efficient algorithm
- Differentiate the fake job recommend by fake sites and be aware from the Scammers



5.3 USER STORIES :

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

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my data by login	High	Sprint-1
	Dashboard	USN-6	As a user , I can view the dashboard and by products		High	Sprit -2
Customer (Web user)	Registration / Login	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard		Sprint -1
Customer Care Executive	Contact with Customers	USN-8	As a Customer customers care executive, I solve the customer Requirements and feedback	I can receive calls from customers	High	Sprint-1
Administrator	Check stock and Price , orders	USN_9	As a Administrator , I can Check the database And stock details and buying and selling prices	I am the administrator of the company	High	Sprint -2

6. PROJECT PLANNING & SCHEDULE

6.1 SPRINT PLANNING & ESTIMATION:

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

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

6.2 SPRINT DELIVERY SCHEDULE:

SPRINT	TASK	MEMBERS
SPRINT 1	Create Registration page , login page , Job search portal , job apply portal in flask	Thabitha Elammathi Dhiksha Emma
SPRINT 2	Connect application to ibm db2	Thabitha Elammathi Dhiksha Emma
SPRINT 3	Integrate ibm Watson assisstant	Thabitha Elammathi Dhiksha Emma
SPRINT 4	Containerize the app and Deploy the application in ibm cloud	Thabitha Elammathi Dhiksha Emma

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date(Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

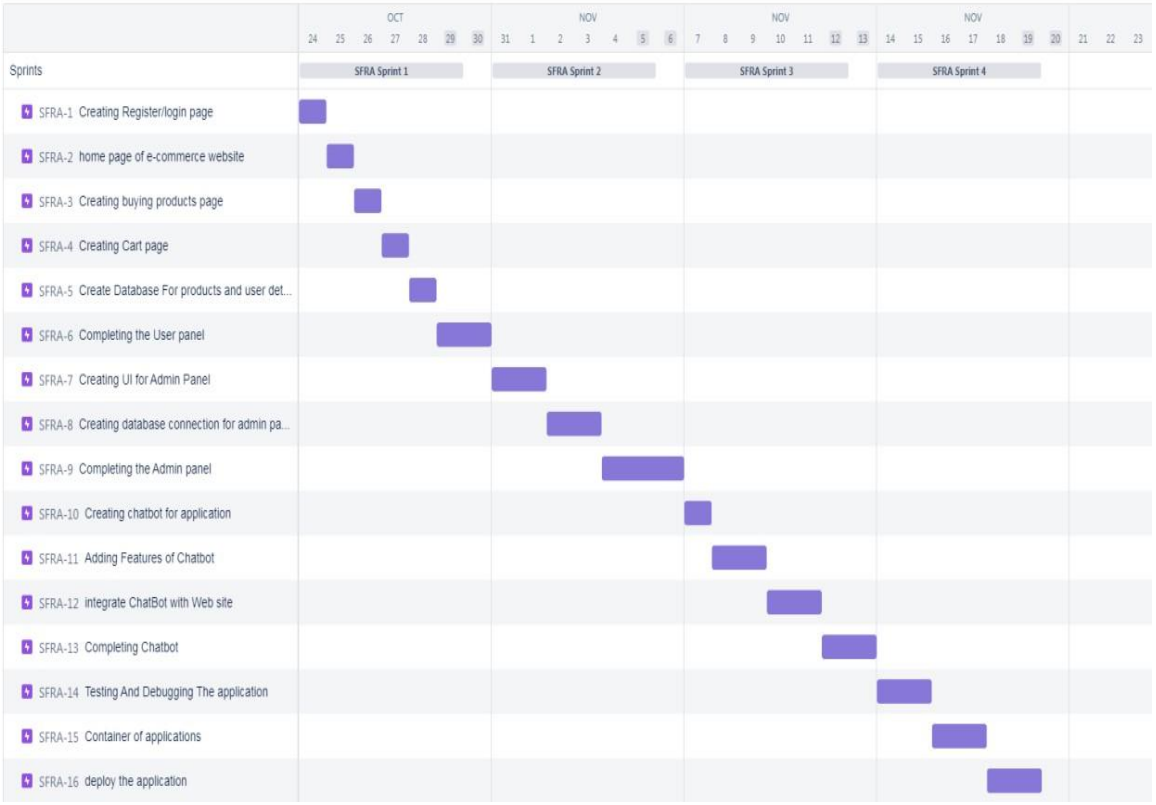
Velocity:

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

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

6.3 REPORTS FROM JIRA:

Burndown Chart:



7. CODING & SOLUTIONING

7.1 FEATURE-1:

HOMEPAGE.HTML:

```
<html>
   <head>
      <title> FASHION FREAK </title>

   </head>
   <style>
      *{
   margin: 0;
   padding: 0;
   font-family: "Times New Roman", Times, serif;

}
.main{
   width: 100%;
   background: linear-gradient(to top,rgba(0,0,0,0.5),rgba(0,0,0,0.5)50%);
   background-position: center;
   background-size: cover;
   height: 100%;
   font-family: "Times New Roman", Times, serif;



}

.navbar{
   width: 100%;
   height:  75px;
   margin: auto;
}
```

```
.icon{
    width: 200px;
    float: left;
    height: 70px;
}

.logo{
    color:#FFFFFF;
    font-size: 35px;

    padding-left: 20px;
    float: left;
    padding-top: 10px;
}

.menu{
    width: 400px;
    float: left;
    height: 70px;
}

ul{
   float: left;
   display: flex;
   justify-content: center;
   align-items: center;
}

ul li{
    list-style: none;
    margin-left: 62px;
    margin-top: 27px;
    font-size: 15px;
```

```
}

ul li a{
    text-decoration: none;
    color: #FFFFFF;

    font-weight: bold;
    transition: 0.4s ease-in-out;
}

ul li a:hover{
    color: rgb(98, 246, 152);

}

.search{
    width: 330px;
    float: left;
    margin-left: 270px;
}

.srch{

    width: 200px;
    height: 40px;
    background: transparent;
    border: 1px solid rgb(98, 246, 152);
    margin-top: 13px;
    color: #FFFFFF;
    border-right: none;
    font-size: 16px;
    float: left;
    padding: 10px;
```

```
    border-bottom-left-radius: 5px;
    border-top-left-radius: 5px;
}

.btn{
    width: 100px;
    height: 40px;
    background:rgb(98, 246, 152) ;
    border: 2px solid rgb(98, 246, 152);
    margin-top: 13px;
    color: #FFFFFF;
    font-size: 15px;
    border-bottom-right-radius: 5px;
    border-bottom-right-radius: 5px;
}

.btn:focus{
    outline: none;
}

.srch:focus{
    outline: none;
}

.content{
    width: 1200px;
    height: auto;
    margin: auto;
    color: #800080;
    position: relative;
}

.content.par{
```

```
    padding-left: 20px;
    padding-bottom: 25px;

    letter-spacing: 1.2px;
    line-height: 30px;
}

.content h1{

    font-size: 50px;
    padding-left: 20px;
    margin-top: 9%;
    letter-spacing: 2px;
}

.content .cn{
    width: 160px;
    height: 40px;
    background: rgb(98, 246, 152);
    border: none;
    margin-bottom: 10px;
    margin-left: 20px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    transition: .4s ease;
}

.content .cn a{
    text-decoration: none;
    color: #FBE7A1;
    transition: .3s ease;
}
```

```
.cn:hover{
    background-color: #FBE7A1;
}

.content span{
    color:rgb(98, 246, 152);
    font-size: 60px;
}

.form{
    width: 250px;
    height: 380px;
    background: linear-gradient(to top,hsla(89, 43%, 51%, 0.3));
    position: absolute;
    top: -20px;
    left: 870px;
    border-radius: 10px;
    padding: 25px;
}

.form h2{
    width: 220px;

    text-align: center;
    color:rgb(98, 246, 152);
    font-size: 22px;

    border-radius: 10px;
    margin: 2px;
    padding: 8px;
}
```

```
.form input{
    width: 240px;
    height: 35px;
    background: rgba(0, 255, 0, 0.5);
}

.form input{
    width: 240px;
    height: 35px;
    background: rgba(0, 255, 0, 0.5);
    border-bottom: 1px solid rgb(98, 246, 152);
    border-top: none;
    border-right: none;
    border-left: none;
    color: #fff;
    font-size: 15px;
    letter-spacing: 1px;
    margin-top: 30px;

}

.form input:focus{
    outline: none;
}

::placeholder{
    color: #fff;

}

.btnn{
    width: 240px;
    height: 40px;
```

```
    background: rgb(98, 246, 152);
    border: none;
    margin-top: 30px;
    font-size: 18px;
    border-radius: 10px;
    cursor: pointer;
    color: #fff;
    transition: 0.4s ease;
}

.btnn:hover{
    background: #fff;
    color: rgb(98, 246, 152);
}

.btnn a{
    text-decoration: none;
    color: #000;
    font-weight: bold;
}

.form .link{

    font-size: 17px;
    padding-top: 20px;
    text-align: center;
}

.form .link a{
    text-decoration: none;
    color: rgb(98, 246, 152);
}
```

```
.liw{
    padding-top: 15px;
    padding-bottom: 10px;
    text-align: center;
}
    </style>
    <body>

        <div class="main">
            <div class="navbar">
                <div class="icon">
                    <h2 class="logo">FASHIONZZ</h2>
                </div>

                <div class="menu">
                    <ul>
                        <li><a href="#">HOME</a></li>
                        <li><a href="#">ABOUT</a></li>

                    </ul>
                </div>

                <div class="search">
                    <input class="srch" type="search" name="" placeholder="TYPE TO SEARCH">
                    <a href="#"><button class="btn">SEARCH</button></a>
                </div>
                        <img src="C:\Users\HP\Downloads\fashion-banner-people-cartoon-characters-flat-
vector-illustration-isolated-fashion-banner-people-cartoon-characters-190292392.jpg">

            </div>


        </div>
```

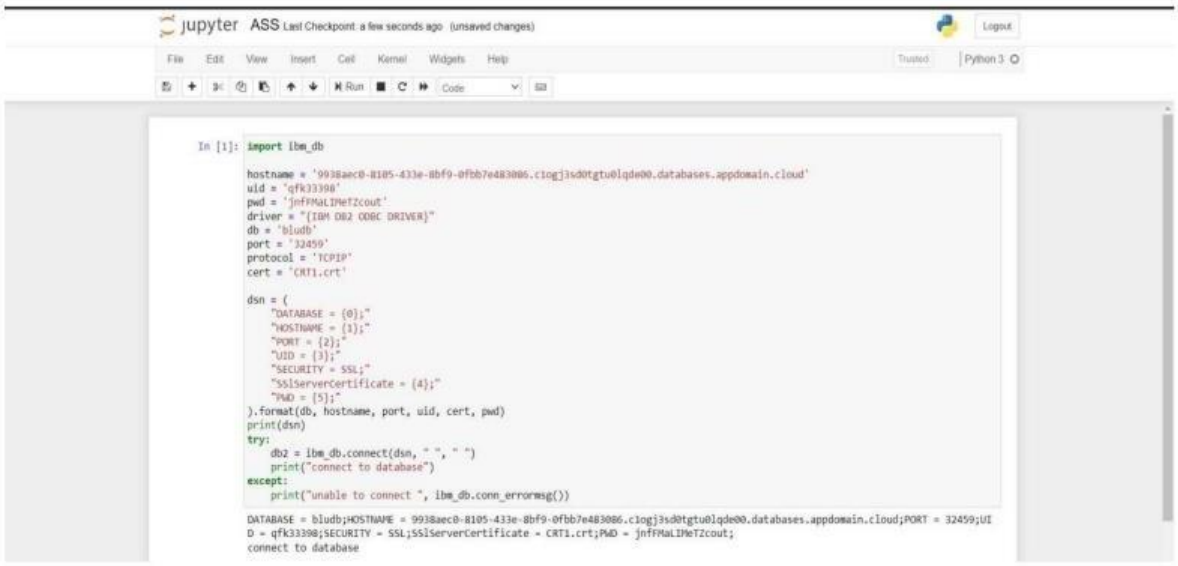
```
    </body>
</html>
```

INTEGRATING CHATBOT WITH HTML PAGE (SOURCE CODE) :

```
<script>

 window.watsonAssistantChatOptions =

{

     integrationID:
     "614a4315-ff80-4187-
     8fe4-2fd9b506b723", //
     The ID of this
     integration.region: "au-
     syd", // The region your
     integration is hosted in.
     serviceInstanceID: "9670dcf8-
     789f-4609-8d7a-6e25c412a9ec",
```

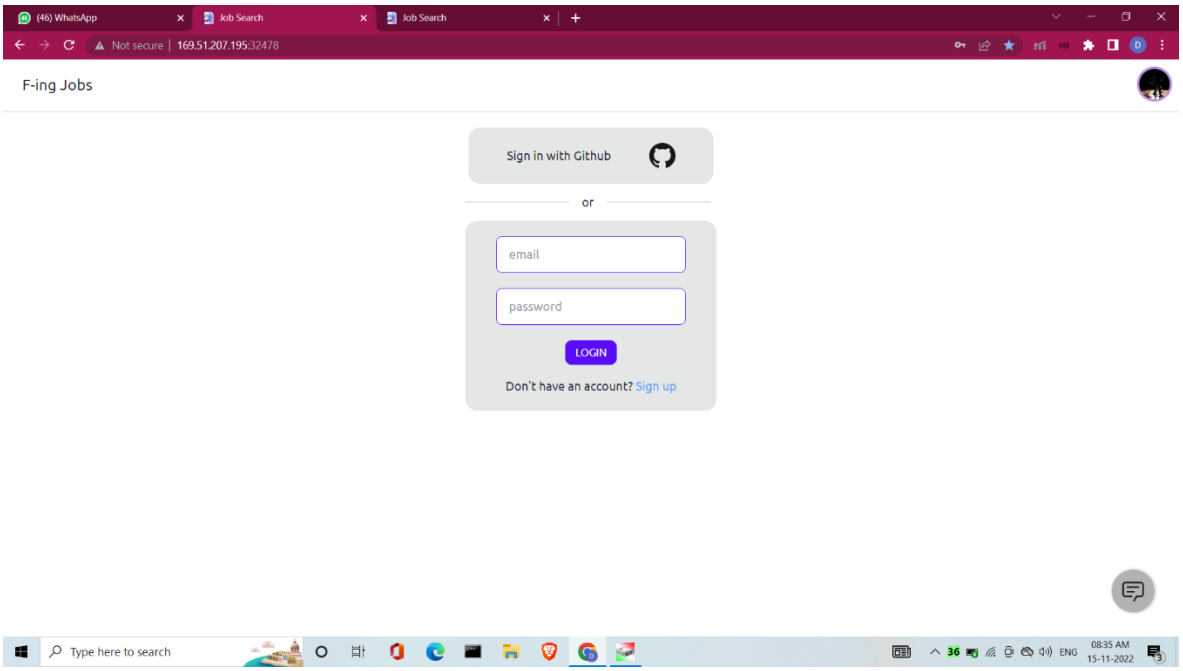
```
    // The ID of your service instance.
    onLoad: function(instance) {
    instance.render(); }
   };
   setTimeout(function(){
    const t=document.createElement('script');
    t.src="https://web-
  chat.global.assistant.watson.appdomain.cloud/versions/" +
  (window.watsonAssistantChatOptions.clientVersion || 'latest') +
  "/WatsonAssistantChatEntry.js";
    document.head.appendChild(t);
   });
  </script>
```

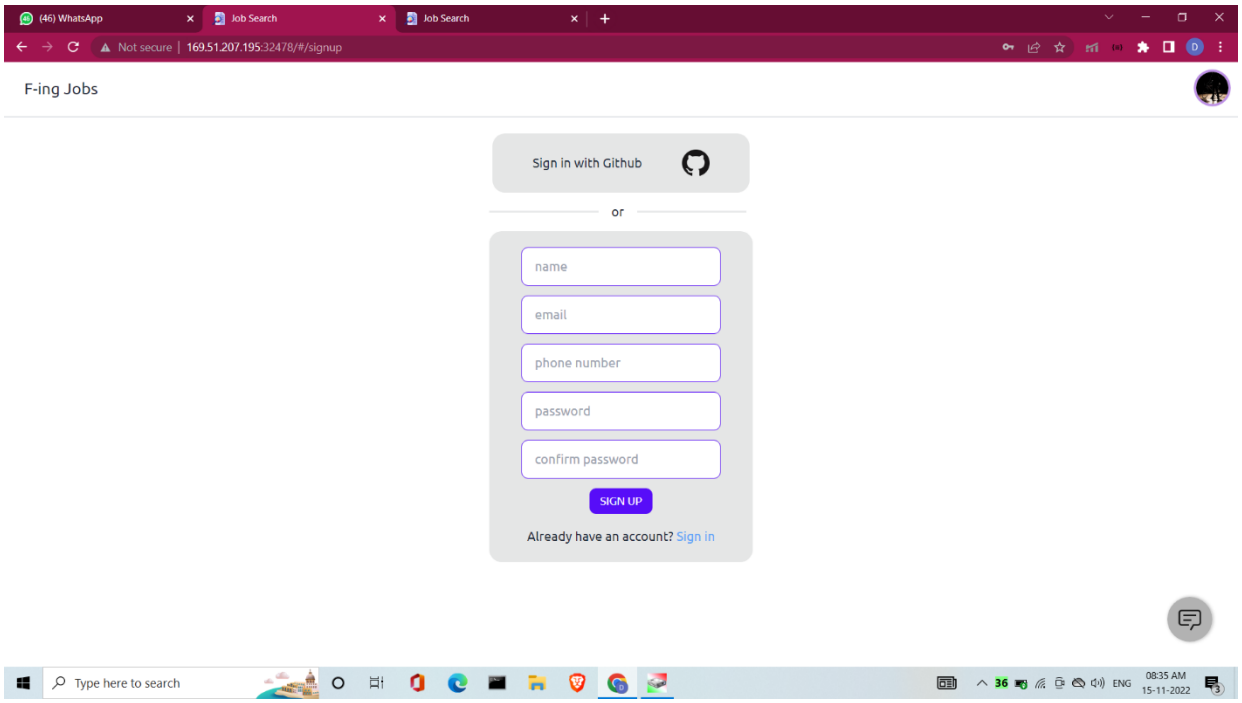
7.2 DATABASE SCHEMA:

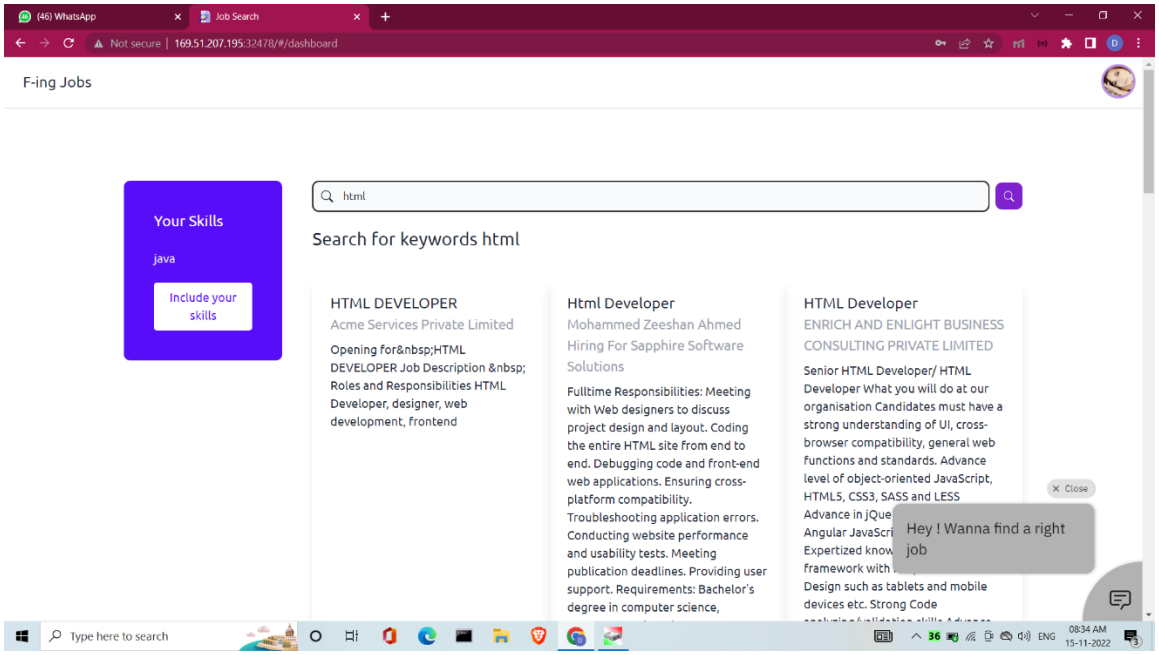


8. TESTING

8.1 TEST CASES:






(46) WhatsApp
Job Search
Not secure | 169.51.207.195:32478/#/dashboard
F-ing Jobs
Your Skills
java
Include your skills
html
Search for keywords html
HTML DEVELOPER
Acme Services Private Limited
Opening for HTML DEVELOPER Job Description   Roles and Responsibilities HTML Developer, designer, web development, frontend
Html Developer
Mohammed Zeeshan Ahmed Hiring For Sapphire Software Solutions
Fulltime Responsibilities: Meeting with Web designers to discuss project design and layout. Coding the entire HTML site from end to end. Debugging code and front-end web applications. Ensuring cross-platform compatibility. Troubleshooting application errors. Conducting website performance and usability tests. Meeting publication deadlines. Providing user support. Requirements: Bachelor's degree in computer science,
HTML Developer
ENRICH AND ENLIGHT BUSINESS CONSULTING PRIVATE LIMITED
Senior HTML Developer/ HTML Developer What you will do at our organisation Candidates must have a strong understanding of UI, cross-browser compatibility, general web functions and standards. Advance level of object-oriented JavaScript, HTML5, CSS3, SASS and LESS
Close
Hey ! Wanna find a right job
Type here to search
ENG
08:34 AM
15-11-2022

9. RESULTS

9.1 PERFORMANCE METRICS:

The performance of a recommendation algorithm is evaluated by using some specific metrics that indicate the accuracy of the system. The type of metric used depends on the type of filtering technique. Root Mean Square Error (RMSE), Receiver Operating Characteristics (ROC), Area Under Cover (AUC), Precision, Recall and F1 score is generally used to evaluate the performance or accuracy of the recommendation algorithms.

Root-mean square error (RMSE). RMSE is widely used in evaluating and comparing the performance of a recommendation system model compared to other models. A lower RMSE value indicates higher performance by the recommendation model. RMSE, as mentioned by [61], can be as represented as follows:

$$RMSE = \sqrt{\frac{1}{N_p}\sum_{u,i}(p_{ui} - r_{ui})^2} \quad (1)$$

where, N_p is the total number of predictions, p_{ui} is the predicted rating that a user u will select an item i and r_{ui} is the real rating.

Precision. Precision can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of recommendations provided, which can be as represented as follows:

$$Precision = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Positive\ (FP)} \quad (2)$$

It is also defined as the ratio of the number of relevant recommended items to the number of recommended items expressed as percentages.

Recall. Recall can be defined as the fraction of correct recommendations or predictions (known as True Positive) to the total number of correct relevant recommendations provided, which can be as represented as follows:

$$Recall = \frac{True\ Positive\ (TP)}{True\ Positive\ (TP) + False\ Negative\ (FN)} \quad (3)$$

It is also defined as the ratio of the number of relevant recommended items to the total number of relevant items expressed as percentages.

F1 Score. F1 score is an indicator of the accuracy of the model and ranges from 0 to 1, where a value close to 1 represents higher recommendation or prediction accuracy. It represents precision and recall as a single metric and can be as represented as follows:

$$F1\ score = 2 \times \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

Coverage. Coverage is used to measure the percentage of items which are recommended by the algorithm among all of the items.

Accuracy. Accuracy can be defined as the ratio of the number of total correct recommendations to the total recommendations provided, which can be as represented as follows:

$$Accuracy = \frac{TP + FN}{TP + FN + TN + FP} \quad (5)$$

Intersection over union (IoU). It represents the accuracy of an object detector used on a specific dataset [62].

$$IoU = \frac{TP}{TP + FN + FP} \tag{6}$$

ROC. ROC curve is used to conduct a comprehensive assessment of the algorithm's performance [57].

AUC. AUC measures the performance of recommendation and its baselines as well as the quality of the ranking based on pairwise comparisons [5].

Rank aware top-N metrics. The rank aware top-N recommendation metric finds some of the interesting and unknown items that are presumed to be most attractive to a user [63]. Mean reciprocal rank (MRR), mean average precision (MAP) and normalized discounted cumulative gain (NDCG) are three most popular rank aware metrics.

MRR. MRR is calculated as a mean of the reciprocal of the position or rank of first relevant recommendation [64][65]. MRR as mentioned by [64][65] can be expressed as follows:

$$MRR = \frac{1}{N_u} \sum_{u \in N_u} \frac{1}{L_u^n [k] \in R_u} \tag{7}$$

where u, N_u and R_u indicate specific user, total number of users and the set of items rated by the user, respectively. L indicates list of ranking length (n) for user (u) and k represents the position of the item found in the he lists L.

MAP: MAP is calculated by determining the mean of average precision at the points where relevant products or items are found. MAP as mentioned by [65] can be expressed as follows.

$$MAP = \frac{1}{N_u |R_u|} \sum_{k=1}^{n} \mathbb{1}(L_u^n [k] \in R_u) P_u @k \tag{8}$$

where P_u represents precision in selecting relevant item for the user.

NDCG: NDCG is calculated by determining the graded relevance and positional information of the recommended items, which can be expressed as follows [65].

$$NDCG_u = \frac{\sum_{k=1}^{n} G(u, n, k) D(k)}{\sum_{k=1}^{n} G^*(u, n, k) D(k)} \tag{9}$$

where D (k) is a discounting function, G (u, n, k) is the gain obtained recommending an item found at k-th position from the list L and G^* (u, n, k) is the gain related to k-th item in the ideal ranking of n size for u user.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

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

DISADVANTAGES:

- Privacy concerns.
- Too many choices.
- Cold-start problem.
- It is costly.
- Uninterrupted internet connection is required for smooth functioning of application.

11. CONCLUSION

we have used ibm cloud services like db2, cloud registry , kubernetes , Watson assistant to create this application , which will be very usefull for candidates who are searching for job and as well as for the company to select the right candidate for their organization

12. FUTURE SCOPE

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. We can use machine learning technicques to recommend data in a efficient way.