**SKILL/JOB RECOMMENDER APPLICATION**

**NALAIYA THIRAN PROJECT BASED LEARNING**

**On**

**PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP**

## A PROJECT REPORT

BINDHIYA J 720819205007

KAMALESH K 720819205018

KARAN P 720819205019

NAWIN S M 720819205028

**BACHELOR OF TECHNOLOGY**

**IN INFORMATION TECHNOLOGY**

## HINDUSTHAN COLLEGE OF ENGINEERING AND TECHOLOGY

Approved by AICTE, New Delhi, Accredited with ‘A’ Grade by NAAC

**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**COIMBATORE – 641 032**

November 2022

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
| **1.** | **INTRODUCTION**  Project Overview  1.2 Purpose | **4** |
| **2.** | **LITERATURE SURVEY**  2.1 Existing problem  2.2 References  2.3 Problem Statement Definition | **5** |
| **3.** | **IDEATION & PROPOSED SOLUTION**  3.1 Empathy Map Canvas  3.2 Ideation & Brainstorming  3.3 Proposed Solution  3.4 Problem Solution fit | **9** |
| **4.** | **REQUIREMENT ANALYSIS**  4.1 Functional requirement  4.2 Non-Functional requirements | **17** |
| **5.** | **PROJECT DESIGN**  5.1 Data Flow Diagrams  5.2 Solution & Technical Architecture  5.3 User Stories | **19** |
| **6.** | **PROJECT PLANNING & SCHEDULING**  6.1 Sprint Planning & Estimation  6.2 Sprint Delivery Schedule  6.3 Reports from JIRA | **22** |
| **7.** | **CODING & SOLUTIONING**  **(Explain the features added in the project along with**  **code)**  7.1 Feature 1  7.2 Feature 2 | **26** |
| **8.** | **TESTING**  8.1 Test Cases  8.2 User Acceptance Testing | **39** |
| **9.** | **RESULTS**  9.1 Performance Metrics | **42** |
| **10.** | **ADVANTAGES & DISADVANTAGES** | **46** |
| **11.** | **CONCLUSION** | **47** |
| **12.** | **FUTURE SCOPE** | **48** |
| **13.** | **APPENDIX**  13.1 Source Code  13.2 GitHub & Project Demo Link | **48** |

## INTRODUCTION

## 1.1 Project Overview

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.

## https://lh3.googleusercontent.com/1OWTBsvpOXh0YVOalvRAGG8uDOBJea7NpyXg5hSSHb61IRRFHTY8txceIQfcIsc9b9coajOEraPoPIAVr5SOr0WFF0iQKVHnHOXk-wAn6XwNjuZFSsdGwreGV7Y10Q

## 1.2 Purpose:

## 

* Aim is to come up with a job recommender system, which takes the skills from LinkedIn and jobs from Indeed and throws the best jobs available for you according to your skills.
* The better a website can match these jobs to the respective seeker, the better the chances of a conversion, and the more popular will the website become through referrals and word of mouth.
* It helps the users to get personalized recommendations, helps users to take correct decisions in their job selection.
* The recommendation system, or recommender system guides users in making predictions for the right jobs relevant to their skills and qualifications.
* To serve the constant cycle of the hiring process in the job applicant’s perspective, here is

a job seeker that looks up for a job relevant to him.

1. **Literature Survey**

**2.1 Existing Problem**

When the whole world is coming back on its feet, those businesses affected by this pandemic disease slowly tries to gain back the momentum it lost. Now is the time when the companies or businesses seek to invest in human resources, which would help them to gain the momentum it lost during this period. When the governments across the world ask businesses to halt the operation in the effort of controlling the pandemic, many companies asked their employees to work remotely. In contrast, many other companies started to reduce their operational cost by terminating employees who were in permanent and contract roles. Individuals who lost their job to the consequence of shutdown are awaiting for their next opportunity. Naturally, we human tries to strive through all difficulties to serve the purpose of our life.Job applicants have become more persistent and proactive in searching for new opportunities that fit their skills. However, companies that are targeting these job seekers are finding it challenging to identify the job seeker’s skill and provide personalized job recommendations

## 2.2 References

1.Title: Toward the next generation of recruitment tools: An online social network- based job recommender system

Author : M Diaby,E Viennet,and T Launay

Abstract : This paper presents a content-based recommender system which proposes jobs to Facebook and LinkedIn users. A variant of this recommender system is currently used by Work4, a San Francisco-based software company that offers Facebook recruitment solutions.Work4 is the world leader in social recruitment technology; to use its applications, Facebook or LinkedIn users explicitly grant access to some parts of their data, and they are presented with the jobs whose descriptions are matching their profiles the most.The profile of a user contains two types of data: interactions data (user's own data) and social connections data (user's friends data). Furthermore the users profiles and the description of jobs are divided into several parts called fields.Our experiments suggest that to predict the users interests for jobs, using basic similarity measures together with their interactions data collected by Work4 can be improved upon.The second part of this study presents a method to estimate the importance of each field of users and jobs in the task of job recommendation.Finally, the third part is devoted to the use of a machine learning algorithm in order to improve the results obtained with similarity measures: we trained a linear SVM (Support Vector Machines). Our results show that using this supervised learning procedure increases the performance of our content-based recommender system.

2.Title : Matching resumes and jobs based on relevance models

Author : Xing Yi,James Allan and W.Bruce Croft

Abstract : We investigate the difficult problem of matching semi-structured resumes and jobs in a large scale real-world collection. We compare standard approaches to Structured Relevance Models (SRM), an extension of relevance-based language model for modeling and retrieving semi-structured documents. Preliminary experiments show that the SRM approach achieved promising performance and performed better than typical unstructured relevance models.

3.Title: Collaborative filtering based online recommendation systems

Author:Basit Mehmood Khan et.al..

Abstract: In recent years, the volume of data present online has grown exponentially. A major portion of this data is related to internet-based e-commerce platforms. The evaluation of such data and/or the extraction of information is difficult due to its huge volume. It is cumbersome for an individual or an organization to obtain the desired results in a timely manner. Recommender Systems (RS) present an automated and efficient solution to this problem. Recommender systems analyze the user profile/behavior and presents products relative to the users' interests. RS maybe based on collaborative filtering, content-based or a hybrid of these techniques. Online recommendation through Collaborative Filtering (CF) plays a vital role in e-commerce and is regarded as one of the best techniques for making possible recommendations for customers. This research analyzes the recommendation systems based on collaborative filtering. Two techniques applied in recommendation system based on collaborative filtering are item-based and user-based approaches. In todays' world, these techniques take place in a global internet environment to produce accurate results according to the need of the user. This paper presents a survey of various state of the art techniques for recommendation systems and highlights the best techniques to generate accurate results.

**5**.Title : Job Recommendation System Using Profile Matching and Web- Crawling

Author : Deepali V Musale et.al.

Abstract : The developed system is job recommendation system for campus recruitment which helps college placement office to match company’s profiles and student’s profiles with higher precision and lower cost. For profile matching, two matching methods are used: semantic matching, tree-based knowledge matching and query matching. These methods are integrated according to representations of attributes of students and companies, and then the profile similarity degree is acquired. Based on profile similarity degree, preference lists of companies and students are generated. Also students can perform keyword based search for job profiles from various job recruitment sites (e.g. Naukari.com,indeed.com). For obtaining data from online recruitment sites system uses web crawling. With loop matching, matching results would be further optimized and provide more effective guidance for recommendation.

## 2.3 Problem Statement Definition

## 

|  |  |
| --- | --- |
| Date | 19 September 2022 |
| Team ID | PNT2022TMID10587 |
| Project Name | Project – Skill/Job Recommender Application |
| Maximum Marks | 2 Marks |

**Customer Problem Statement Template:**

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 recommend job vacancies that fit properly to the jobseekers profiles. Thus, the contributions of this work are threefold, made publicly available a new dataset formed by a set of job seekers profiles and a set of job vacancies collected from different job search engine sites, put forward the proposal of a framework for job recommendation based on professional skills of job seekers, and carried out an evaluation to quantify empirically the recommendation abilities of two state-of-the-art methods, considering different configurations, within the proposed framework. Thus present a general panorama of job recommendation task aiming to facilitate research and real-world application design regarding this important issue. Job matching, job seeking, job search, job recommender systems.

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

1. **IDEATION & PROPOSED SOLUTION**

**3.1 Emapthy Map Canvas**

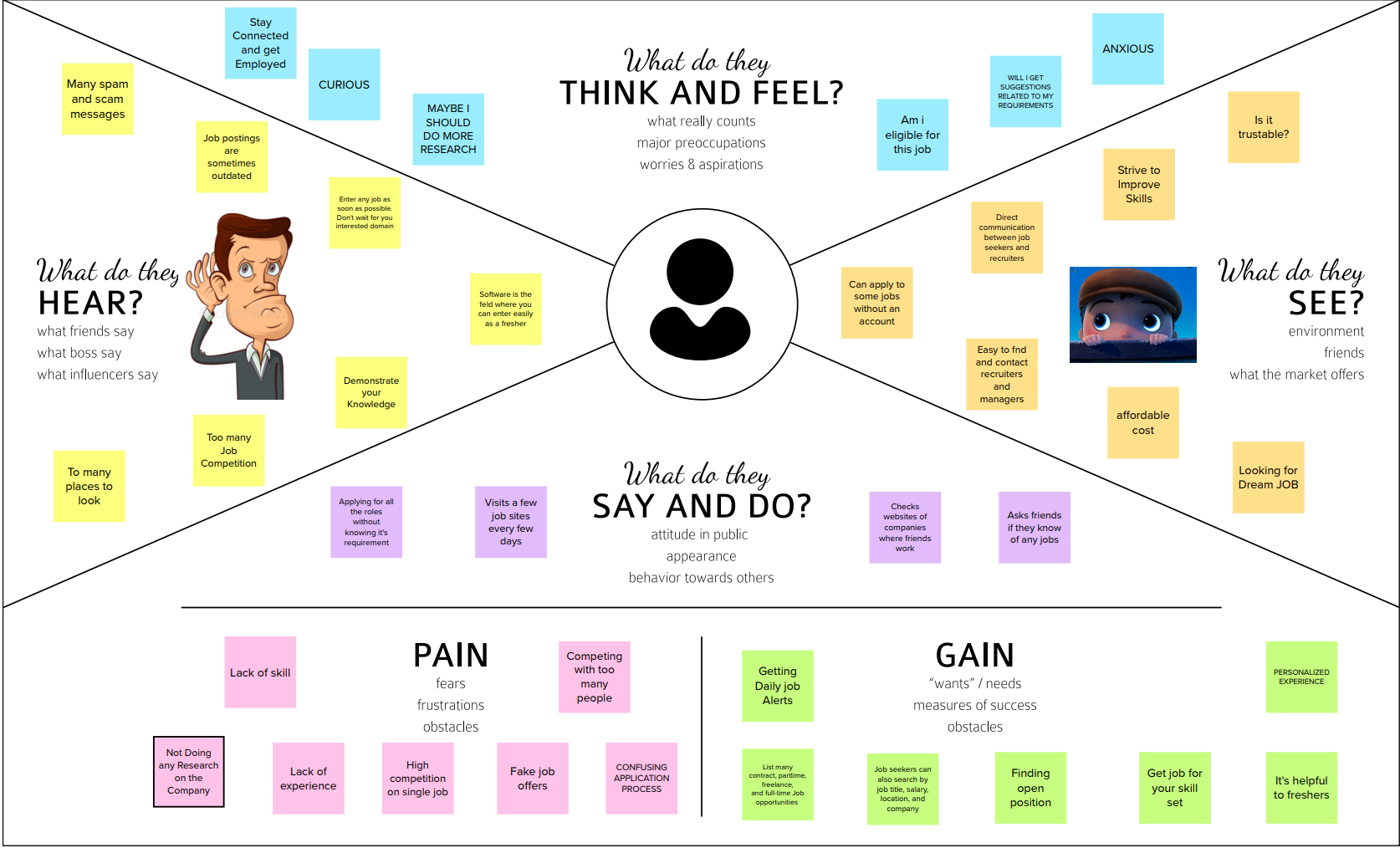
|  |  |
| --- | --- |
| Date | 19 September 2022 |
| Team ID | PNT2022TMID10587 |
| Project Name | Project – Skill/Job Recommender Application |
| Maximum Marks | 4 Marks |

**Empathy Map Canvas: Crude Oil Price Prediction**

The Skill/Job Recommender Application empathy map shows the visual that captures knowledge about a user’s behaviors and attitudes. It is more useful for the teams better understand their users.

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user’s behaviors and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user’s perspective along with his or her goals and challenges.

**The below Empathy Map shows the user’s behaviors and attitudes of Skill/Job Recommendation system:**



**3.2 Ideation & Brainstorming**

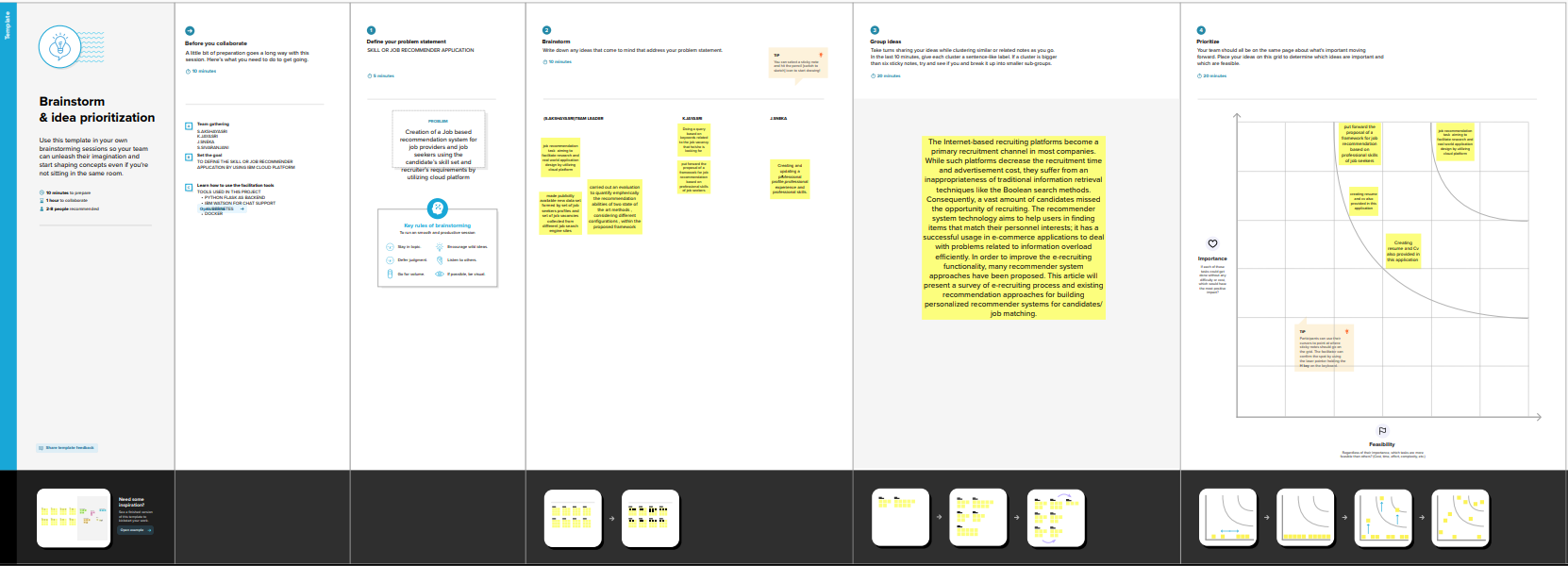
|  |  |
| --- | --- |
| Date | 19 September 2022 |
| Team ID | PNT2022TMID10587 |
| Project Name | Project – Skill/Job Recommender Application |
| Maximum Marks | 4 Marks |

#### Brainstorm & Idea Prioritization Template:

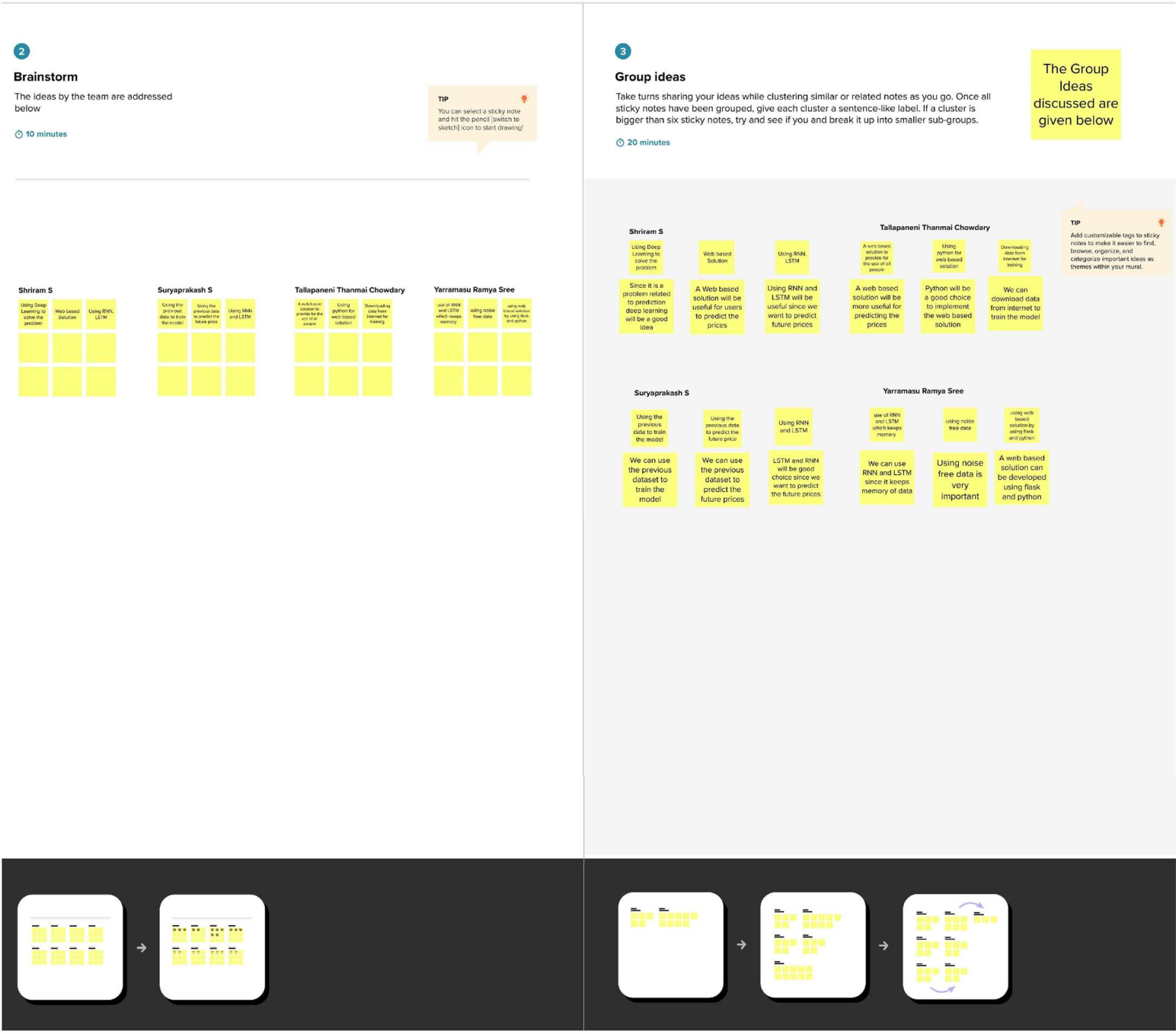
For Skill/Job Recommender Application, Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants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

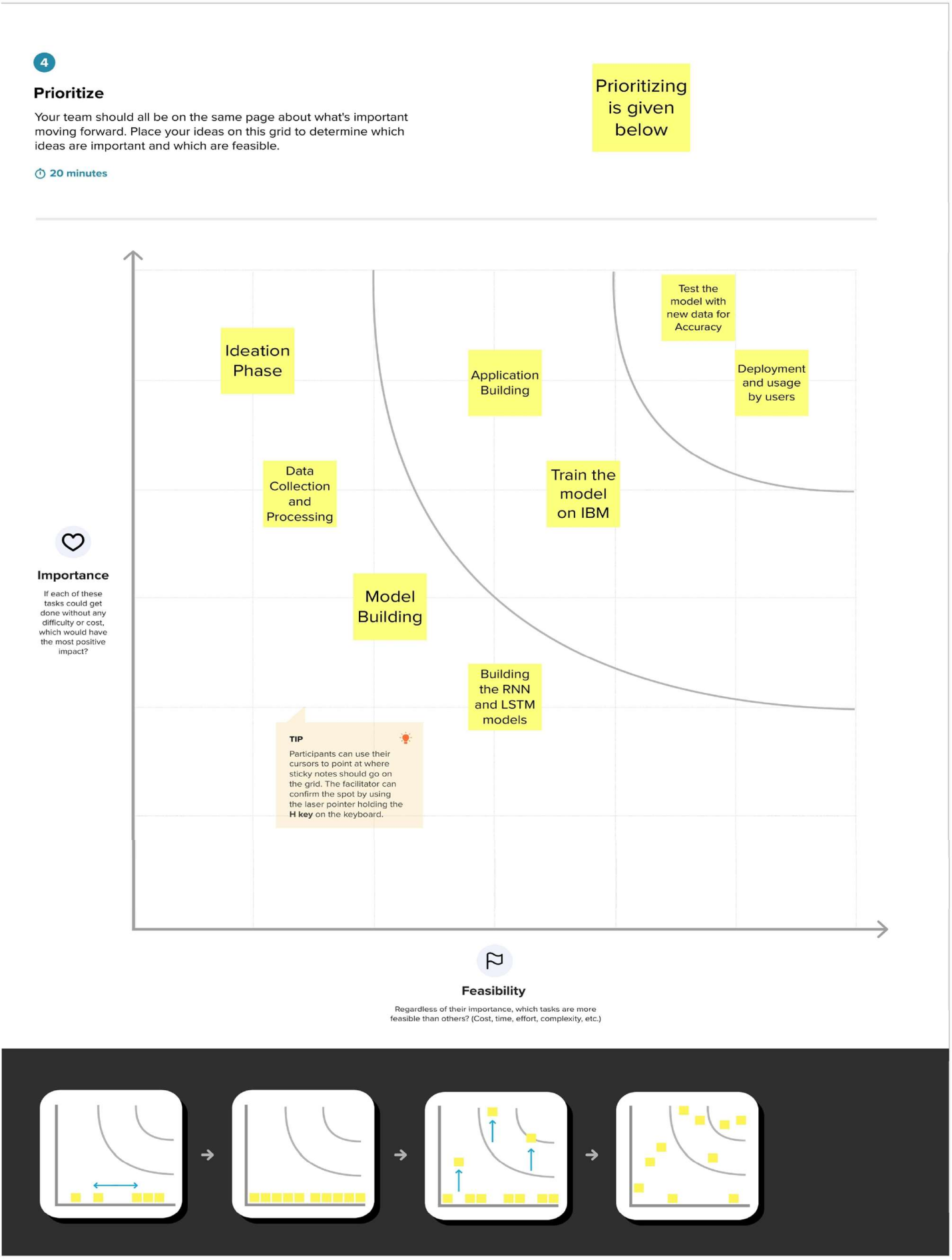
We as a team conducted Brainstorming which provided many solutions that leaded to problem solving.

#### 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**

|  |  |
| --- | --- |
| Date | 19 September 2022 |
| Team ID | PNT2022TMID10587 |
| Project Name | Project – Skill/Job Recommender Application |
| Maximum Marks | 2 Marks |

**Proposed Solution Template:**

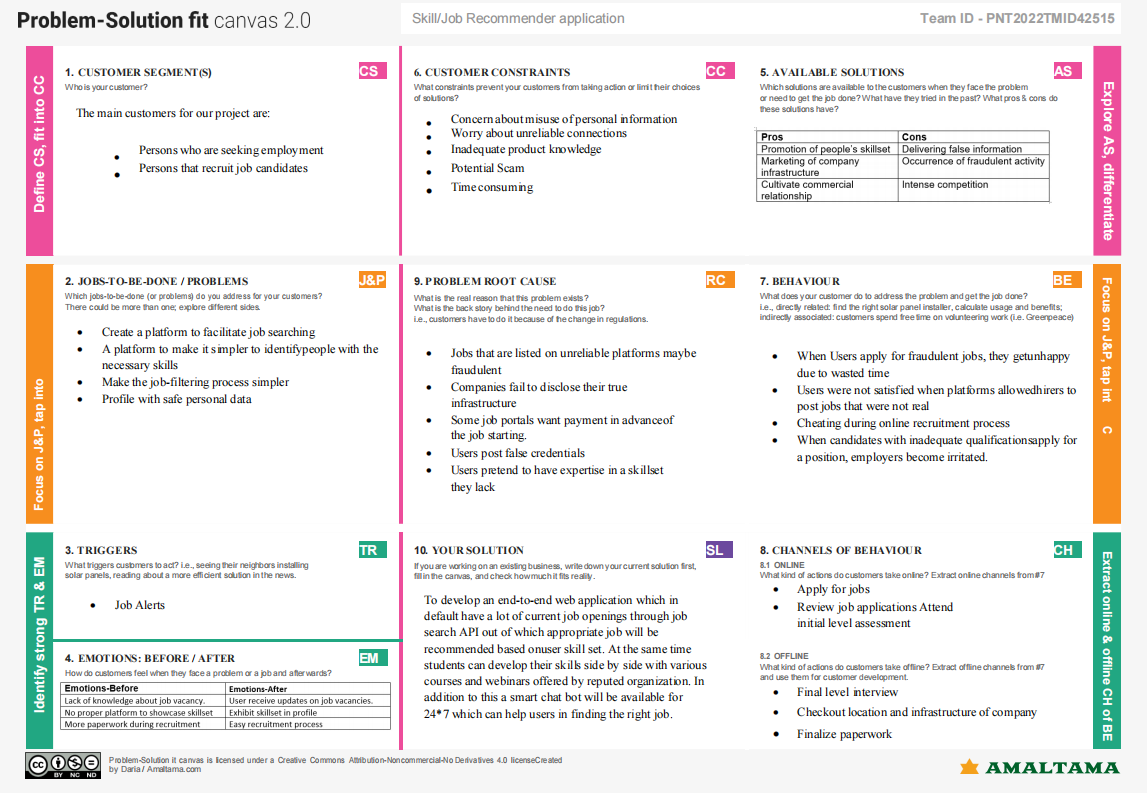
Project team shall fill the following information in proposed solution template.

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Problem Statement (Problem to be solved) | * Look for field based jobs as searching for fields as a whole is time-consuming. * Estimating salaries based on technical skills. Premium policy is an issue to the users. |
| 2. | Idea / Solution description | * Free access to every users. * Filtering job by it’s categories.   ○ Salary calculator for estimation of pay. |
| 3. | Novelty / Uniqueness | * Refinement of the job fields. * Earnings estimator based on knowledge of users. |
| 4. | Social Impact / Customer Satisfaction | * Open doors for every users as there is free access. * Users stay up to date of the offers. |
| 5. | Business Model (Revenue Model) | * Advertising about the platform. * Regularly updating new technologies |

|  |  |  |
| --- | --- | --- |
| 6. | Scalability of the Solution | * Scalable at Professional Training and Coaching. * Scalability in finding more parent- friendly environment. * Creating a positive culture is the main cause in maximizing the productivity. |

**3.4 Problem Solution fit**

|  |  |
| --- | --- |
| Date | 19 September 2022 |
| Team ID | PNT2022TMID10587 |
| Project Name | Project – Skill/Job Recommender Application |
| Maximum Marks | 2 Marks |



**4. REQUIREMENT ANALYSIS**

**4.1 Functional Requirements**

|  |  |
| --- | --- |
| Date | 20 October 2022 |
| Team id | PNT2022TMID10587 |
| Project Name | Skill/Job Recommender Application |
| Maximum Marks | 4 Marks |

**Functional Requirements:**

Following are the functional requirements of the proposed solution.

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Registration | Registration via Form And via Gmail |
| FR-2 | User Confirmation | Confirmation through  Email That is through OTP |
| FR-3 | 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. |
| FR-4 | User Login | Login through Form Login through Gmail |
| FR-5 | User Search | Exploration of Jobs based on job filters and skill recommendations. |
| FR-6 | User Profile | Updation of the user profile through the login credentials |
| FR-7 | User Acceptance | Confirmation of the Job. |

**4.2 Non Functional Requirements**

|  |  |
| --- | --- |
| Date | 20 October 2022 |
| Team id | PNT2022TMID10587 |
| Project Name | Skill/Job Recommender Application |
| Maximum Marks | 4 Marks |

**Non Functional Requirements**

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

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | This application can be used by the job seekers to login and search for the job based on his Skills set. |
| NFR-2 | **Security** | This application is secure with separate login for Job Seekers as well as Job Recruiters. |
| NFR-3 | **Reliability** | This application is open-source and feel free to use, without need to pay anything. The enormous job openings will be provided to all the job seekers without any limitations. |
| NFR-4 | **Performance** | The performance of this application is quicker response and takes lesser time to do any process. |
| NFR-5 | **Availability** | This application provides job offers and recommends Skills for a Particular Job openings. |
| NFR-6 | **Scalability** | The Response time of the application is quite faster compared to any other application. |

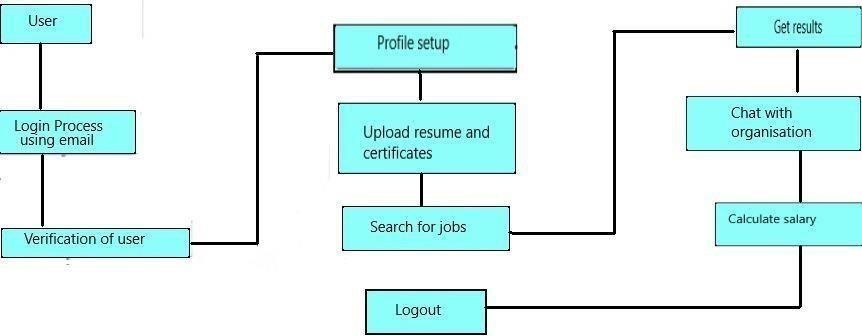
### 5.Project Design

**5.1 Data Flow Diagrams**

|  |  |
| --- | --- |
| Date | 15 October 2022 |
| Team Id | PNT2022TMID10587 |
| Project Name | Project – Skill and Job Recommender |
| Maximum Marks | 4 Marks |

**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 Architechture

**Technology Architecture:**

Project shall full fill the following information in this technology architecture .

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
| 1. | Is the System Robust? | Yes, it is partially buildable platform as the budget required will be more as cloud is a pay per use model and time taken will be quite. |
| 2. | Is it highly modifiable? | Indeed, the framework is modifiable and it can own up to the progressions by recognizing blunders that requirements to be fixed and new functionalities. It is exceptionally receptive to the progressions. |
| 3. | Is it Scalable? | Indeed, the framework proposed is exceptionally versatile as it can deal with the developing responsibility where great execution is likewise expected to effectively work. Organization of the stage has been finished utilizing different OS virtualization stage it will deal with the responsibility genuinely. |

5

### 5.3 User Stories

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional**  **Requirement (Epic)** | **User Story**  **Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| Customer  (Mobile user) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can access my account / dashboard | High | Sprint-1 |
|  |  | USN-2 | As a user, I will receive confirmation email once I have 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. |  | High | Sprint-1 |
|  | Dashboard | USN-5 | As a user, I can access my dashboard after signing in. | I can access my account / dashboard | High | Sprint-1 |
| Customer (Web user) | Access | USN-6 | As a user, I can setup a profile, and basic details by signing in. |  |  |  |
|  |  | USN-7 | As a user, I will upload my resume, certificates, and other requirements. | I can perform several task in the 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 |

### 6.Project Planning & Scheduling

**6.1Sprint Planning & Estimation**

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

Use the below template to create product backlog and sprint schedule

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User**  **Story Number** | **User Story / Task** | **Priority** | **Acceptance criteria** | **Team Members** |
| Sprint-1 | UI Design | USN-1 | As a user, I can see and experience an awesome user interface in the website | Medium | Better Impression about a website | Kamalesh,Nawin |
| Sprint-1 | Registration | USN-2 | As a user, I can register for the application by entering my email, password, and confirming my password. | High | I can access my account *I*  dashboard | Kamalesh, Karan |
| Sprint-1 |  | USN-3 | As a user, I will receive confirmation email once I have registered for the application | High | I can receive confirmation email &  click confirm | Karan,Nawin |
| Sprint-1 |  | USN-4 | As a user, I can register for the application through Facebook | Low | I can register & access the dashboard with Facebook Login | Nawin,Bindhiya |
| Sprint-1 |  | USN-5 | As a user, I can register for the application through Gmail | Medium | I can receive confirmation email &  click confirm | Kamalesh,Karan,  Nawin |
| Sprint-1 | Login | USN-6 | As a user, I can log into the application by entering email & password | High | I can access my account *I*  dashboard | Kamalesh,Nawin |
| Sprint-! | Flask | USN-7 | As a user, I can access the website in a second | High | I can access my account *I*  dashboard | Karan,Nawin |

### 

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Priority** | **Acceptance criteria** | **Team Members** |
| Sprint-1 | Dashboard | USN-8 | As a user, If I Logged in correctly, I can view my dashboard and I can navigate to any pages which are already listed there. | High | I can access all the pages/ dashboard | Karan,Bindhiya |
|  |  |  | Submission Of Sprint-1 |  |  |  |
| Sprint-2 | User Profile | USN-9 | As a user, I can view and update my details | Medium | I can modify my details/data | Kamalesh,Nawin |
| Sprint-2 | Database | USN-10 | As a user, I can store my details and data in the website w | Medium | I can store my data | Bindhiya,Kamalesh |
| Sprint-2 | Cloud Storage | USN-11 | As a user, I can upload my photo, resume and much more in the website. | Medium | I can Upload my documents and details | Karan,Nawin |
| Sprint-2 | Chatbot | USN-12 | As a user, I can ask the Chatbot about latest job openings, which will help me and show the recent job openings based on my profile | High | I can know the recent job openings | Nawin,Karan,  Kamalesh |
| Sprint-2 | Identity-Aware | USN-13 | As a User, I can access my account by entering by correct login credentials. My user credentials is only displayed to me. | High | I can have my account safely | Kamalesh,Nawin,  Bindhiya |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Submission of Sprint-2 |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Priority** | **Acceptance criteria** | **Team Members** |
| Sprint-3 | Sendgrid service | USN-14 | As a user, I can get a notification or mail about a job opening with the help of sendgrid service. | Medium | I can get a notification in a second. | Kamalesh,Bindhiya,  Nawin |
| Sprint-3 | Learning Resource | USN-15 | As a user, I can learn the course and I will attain the skills which will be useful for developing my technical skills. | High | I can gain the knowledge and skills | Kamalesh,Nawin |
| Sprint-3 | Docker | USN-16 | As a user, I can access the website in any device | High | I can access my account in any device | Kamalesh,  Karan |
| Sprint-3 | Kubernates | USN-17 | As a user, I can access the website in any device | High | I can access my account in any device | Nawin,Kamalesh |
| Sprint-3 | Deployment in cloud | USN-18 | As a user, I can access the website in any device | High | I can access my account in any device | Karan,Nawin |
| Sprint-3 | Technical support | USN-19 | As a user, I can get a customer care support from the website which will solve my queries. | Medium | I can tackle my problem & queries. | Bindhiya,  Kamalesh |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Submission of Sprint-3 |  |  |  |
| Sprint-4 | Unit Testing | USN-15 | As a user, I can access the website without any interruption | High | I can access the website without any interruption | Kamalesh,Nawin |
| Sprint-4 | Integration testing | USN-16 | As a user, I can access the website without any interruption | High | I can access the website without any interruption | Kamalesh,Karan |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Priority** | **Acceptance criteria** | **Team Members** |
| Sprint-4 | System testing | USN-17 | As a user, I can access the website without any interruption | High | I can access the website without any interruption | Karan,Nawin |
| Sprint-4 | Correction | USN-18 | As a user, I can access the website without any interruption | High | I can access the website without any interruption | Nawin,Karan,  Kamalesh |
| Sprint-4 | Acceptance testing | USN-19 | As a user, I can access the website without any interruption | High | I can access the website without any interruption | Bindhiya,  Kamalesh |
|  |  |  | Submission of Sprint-4 |  |  |  |

**6.2 Sprint Delivery Schedul**

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint | Total Story Points | Duration | Sprint Start Date | Sprint End Date (Planned) | Story Points Completed (as on Planned End Date) | Sprint Release Date (Actual) |
| Sprint-1 | 20 | 6 Days | 24 Oct 2022 | 30 Oct 2022 | 20 | 30 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 31 Oct 2022 | 06 Nov 2022 | 20 | 06 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 13 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

**6.3 Reports From JIRA**

# Velocity:

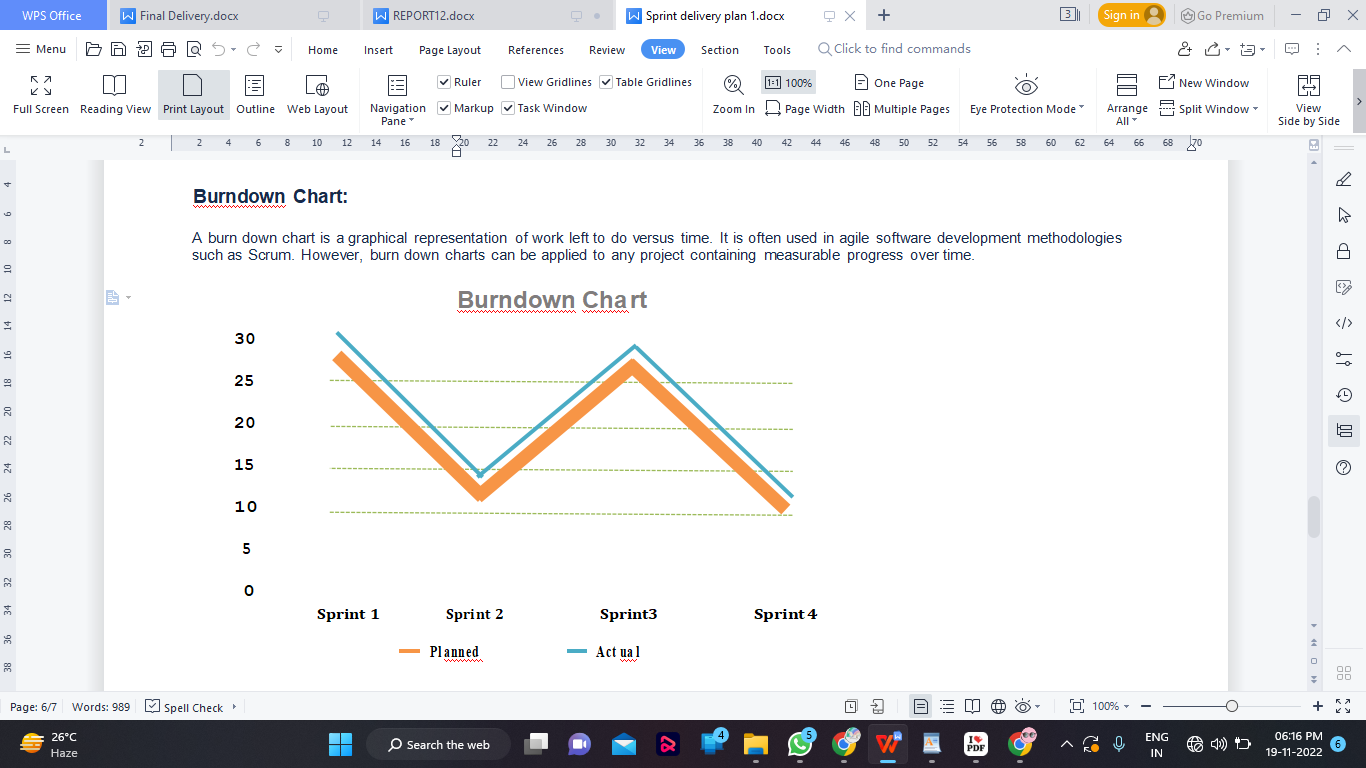
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

*sprint duration* 20

*velocity* 10

# Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.



1. Coding & Solutioning(Explain the features added in the project along with code)
   1. Feature 1

<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>

<style>

.rus{

border-radius: 0%

};

.black{ color:black;

}

</style>

</head>

<body>

<!-------========================================================-->

<nav class="navbar navbar-inverse rus">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Skill and Job Recommender</a>

</div>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-log-in"></span> Log out</a></li>

</ul>

</div>

</nav>

<!-------========================================================-->

<div class="container">

<ul class="nav nav-tabs">

<li class="active"><a data-toggle="tab" href="#home">Profile</a></li>

<li><a data-toggle="tab" href="#menu1">Eligibility Test</a></li>

<li><a data-toggle="tab" href="#menu2">Find Job</a></li>

</ul>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<h3>PROFILE</h3>

<hr>

<form>

<div class="alert alert-danger">

<div class="form-group">

<label for="name"><span clss="black">Name: </span></label>

<input type="text" class="form-control" id="name" value="Mahadevan">

</div>

<div class="form-group">

<label for="email"><span class="black">Email Address </span></label>

<input type="text" class="form-control" id="email" value="Mahadevan@gmaiil.com">

</div>

<div class="form-group">

<label for="email"><span class="black">Mobile No </span></label>

<input type="text" class="form-control" id="email" value="9003824096">

</div>

<div class="form-group">

<label for="quali"><span class="black">What is your Nationality

</span></label>

<input type="text" class="form-control" id="quali" value="India">

</div>

<div class="form-group">

<label for="quali">What is your Qualification</label>

<input type="text" class="form-control" id="quali" value="BE">

</div>

<div class="form-group">

<label for="quali">What year did you complete your bachelor Degree?</label>

<input type="text" class="form-control" id="quali" value="2023">

</div>

?</label>

<div class="form-group">

<label for="experience">How many years of work experience do you have

<input type="text" class="form-control" id="experience" value="0">

</div>

<div class="form-group">

<label for="place">Where have you graduated high school degree ?</label>

<input type="text" class="form-control" id="place" value="Government high school Ariyalur">

</div>

<div class="form-group">

<label for="SSLC">What is your SSLC Percentage :</label>

<input type="text" class="form-control" id="SSLC" value="90">

</div>

<div class="form-group">

<label for="HSC">What is your HSC Percentage :</label>

<input type="text" class="form-control" id="HSC" value="90">

</div>

<div class="form-group">

<label for="degree">What is your Percentage in bachelor Degree:</label>

<input type="text" class="form-control" id="degree" value="90.5">

</div>

<textarea class="form-control" id="aboutyou" rows="3" placeholder="Write few words about yourself"></textarea> <br> <br>

<button type="submit" class="btn btn-primary btn-success">Save changes</button> <br> <br>

</div>

</form>

</div>

<div id="menu1" class="tab-pane fade">

<h3>Eligibility Test </h3> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio" name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio" name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio" name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio" name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio" name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio" name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio" name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio" name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio" name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio" name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio" name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio" name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio" name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio" name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio" name="switch">switch</label>

</div>

</div><hr>

<div class="alert alert-info">

<label>What function is used to display output on the screen?</label>

<div class="radio">

<label><input type="radio" name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio" name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio" name="switch">switch</label>

</div>

</div>

<button type="submit" class="btn btn-primary btn-success">Submit the test</button> <br> <br>

</div>

<div id="menu2" class="tab-pane fade">

<h3>Find the job</h3> <hr>

<br>

job">

<div class="container">

<! >

<form>

<div class="input-group">

<input type="text" class="form-control" id="myInput" placeholder="Look for a

<div class="input-group-btn">

<button class="btn btn-default" type="submit">

<i class="glyphicon glyphicon-search"></i>

</button>

</div>

</div>

</form> <br> <br>

<! >

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Java Developer</h4>

<h5 class="media-heading"> Company Name : JK Software Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btn-success">Apply </button>

<br> <hr>

</div>

</div>

</div>

<br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">.NET Developer</h4>

<h5 class="media-heading"> Company Name : JK Software Solution</h4>

<br> <hr>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btn-success">Apply </button>

</div>

</div>

</div>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">PHP Developer</h4>

<h5 class="media-heading"> Company Name : JK Software Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btn-success">Apply </button>

<br> <hr>

</div>

</div>

</div>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">UX /UI Designer</h4>

<h5 class="media-heading"> Company Name : JK Software Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btn-success">Apply </button>

<br> <hr>

</div>

</div>

</div>

</div>

<script>

$(document).ready(function(){

$("#myInput").on("keyup", function() { var value = $(this).val().toLowerCase();

$("#myDIV \*").filter(function() {

$(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)

});

});

});

</script>

</body>

</html>

Feature 7.2

Back End Feature 1

from flask import \*#importing flask (Install it using python -m pip install flask) import os

import ibm\_db import bcrypt

#import send\_from\_directory

app = Flask( name ) #initialising flask app.secret\_key = ''

PEOPLE\_FOLDER = os.path.join('static', 'people\_photo')

import ibm\_db

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954- 7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32733;SECURITY=SSL;S SLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45 Raj9A0rr",'','')

@app.route("/") #defining the routes for the home() funtion (Multiple routes can be used as seen here)

@app.route("/home")

def dashboard():

return render\_template("dashboard.html") #rendering our home.html contained within /templates

def createjobseeker\_profile(): if request.method == 'POST':

name = request.form['name'] email = request.form['email']

mobileno=request.form['mobileno'] nationality=request.form['nationality'] qualification=request.form['qualification'] passout=request.form['passout'] experience=request.form['experience"]; degree=request.form['degree']; per\_sslc=request.form['per\_sslc']; per\_hsc=request.form['per\_hsc']; per\_degree=request.form['per\_degree'];

tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD\_DIR'],tum.filename))

hash=bcrypt.hashpw(password.encode('utf-8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"

stmt = ibm\_db.prepare(conn, query) ibm\_db.bind\_param(stmt,1,email) ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt) if not isUser:

insert\_sql = "INSERT INTO JOBSEEKER (name, email, mobileno, nationality, qualification, passout, experience, degree,per\_sslc,per\_hsc,per\_degree) VALUES (?,?,?,?,?,?,?,?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql) ibm\_db.bind\_param(prep\_stmt, 1, name)

ibm\_db.bind\_param(prep\_stmt, 2, email)

ibm\_db.bind\_param(prep\_stmt, 3, mobileno)

ibm\_db.bind\_param(prep\_stmt, 4, nationality)

ibm\_db.bind\_param(prep\_stmt, 5, qualification)

ibm\_db.bind\_param(prep\_stmt, 6, passout)

ibm\_db.bind\_param(prep\_stmt, 7, experience)

ibm\_db.bind\_param(prep\_stmt, 8, degree)

ibm\_db.bind\_param(prep\_stmt, 9, per\_sslc)

ibm\_db.bind\_param(prep\_stmt, 10, per\_hsc)

ibm\_db.bind\_param(prep\_stmt, 11, per\_degree) ibm\_db.execute(prep\_stmt)

return render\_template('dashboard.html')

return render\_template('dashboard.html',name='Home')

@app.route('/logout') def logout():

session.pop('email', None) return redirect(url\_for('login'))

**Feature 2**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial- scale=1">

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst rap.min.css">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m in.js"></script>

<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra p.min.js"></script>

<style>

.rus{

border-radius: 0%

};

.black{ color:black;

}

</style>

</head>

<body>

<nav class="navbar navbar-inverse rus">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Skill and Job Recommender</a>

</div>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-log- in"></span> Log out</a></li>

</ul>

</div>

</nav>

<div class="container">

<ul class="nav nav-tabs">

<li class="active"><a data-toggle="tab" href="#home">Company Profile</a></li>

<li><a data-toggle="tab" href="#menu2">Find candidate</a></li>

</ul>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<h3>COMPANY PROFILE</h3>

<hr>

<form>

<div class="alert alert-danger">

<div class="form-group">

<label for="name"><span clss="black">Company Name:

</span></label>

<input type="text" class="form-control" id="name" value="Kamalesh">

</div>

<div class="form-group"> <label for="email"><span class="black">Company Email Address </span></label>

<input type="text" class="form-control" id="email" value="kamalthekindboy@gmail.com">

</div>

<div class="form-group">

<label for="email"><span class="black">Mobile No

</span></label>

<input type="text" class="form-control" id="email" value="9003824096">

</div>

<div class="form-group">

<label for="quali"><span class="black">Location</span></label>

<input type="text" class="form-control" id="quali" value="India">

</div>

<textarea class="form-control" id="aboutyou" rows="3" placeholder="Write few words about company"></textarea> <br>

<br>

<button type="submit" class="btn btn-primary btn- success">Save changes</button> <br> <br>

</div>

</form>

</div>

<div id="menu2" class="tab-pane fade">

<h3>Find the Condidate</h3> <hr>

<br>

<div class="container">

<!

>

<form>

<div class="input-group">

<input type="text" class="form-control" id="myInput"

placeholder="Look for a candidate">

<div class="input-group-btn">

<button class="btn btn-default"

type="submit"> search"></i>

<i class="glyphicon glyphicon-

</button>

</div>

</div>

</form> <br> <br>

<! >

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Name :

Mahadevan</h4>

<h5 class="media-heading"> Qualification : B.E (Computer Science Engineering )</h4> <button type="submit" class="btn btn-primary btn- success">Know More </button> <br>

<hr>

</div>

</div>

</div>

<br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Name :

Manikandan</h4>

<h5 class="media-heading"> Qualification : B.E (Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btn- success">Know More </button> <br> <hr>

</div>

</div>

</div> <br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div> Karan</h4>

<div class="media-body">

<h4 class="media-heading">Name :

<h5 class="media-heading"> Qualification : B.E

(Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btn- success">Know More </button> <br> <hr>

</div>

</div>

</div> <br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png" class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Name :

Ram</h4>

<h5 class="media-heading"> Qualification : B.E

(Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btn- success">Know More </button> <br> <hr>

</div>

</div>

</div> <br>

</div>

</div>

</div>

<script>

$(document).ready(function(){

$("#myInput").on("keyup", function() { var value = $(this).val().toLowerCase();

$("#myDIV \*").filter(function() {

$(this).toggle($(this).text().toLowerCase().indexOf(value)

> -1)

});

});

});

</script>

</body>

</html>

CHATBOT :

**Chatbot has been implemented to provide**

assistance .

window.watsonAssistantC hatOptions = {

integrationID: "d73273d3-3f44-4304- 84ee-8fd243016d1d", // The ID of this integration.

region: "jp-tok", // The region your integration is hosted in. serviceInstanceID: "81229104-ee6b-46da- ac1c-67ede110663a", // 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.clie ntVersion || 'latest') + "/WatsonAssistantChatEntry.js";

document.head.appendChild(t);

});

7.3 Database Schema(if Applicable) :

import os

from sendgrid import SendGridAPIClient from sendgrid.helpers.mail import Mail

# from\_address we pass to our Mail object, edit with your name FROM\_EMAIL = 'Your\_Name@SendGridTest.com'

def SendEmail(to\_email):

""" Send an email to the provided email addresses

:param to\_email = email to be sent to

:returns API response code

:raises Exception e: raises an exception """ message = Mail(

from\_email=FROM\_EMAIL, to\_emails=to\_email,

subject='A Test from SendGrid!',

html\_content='<strong>Hello there from SendGrid your URL is: ' + '<a href=''https://github.com/cyberjive''>right here!</a></strong>')

try:

sg = SendGridAPIClient(os.environ.get('SENDGRID\_API\_KEY')) response = sg.send(message)

code, body, headers = response.status\_code, response.body, response.headers print(f"Response Code: {code} ")

print(f"Response Body: {body} ") print(f"Response Headers: {headers} ") print("Message Sent!")

except Exception as e: print("Error: {0}".format(e))

return str(response.status\_code) if \_name\_ == "\_main\_":

SendEmail(to\_email=input("Email address to send to? "))

**9 RESULTS**

The project has been completed as we expected.

We ensured that Database was designed and well connected to our project.

The Expected results were gotten.

**10 ADVANTAGES & DISADVANTAGES:**

 Person who looks for a job can easily find a suitable job based on their skill set.

 Person can check their eligibility by attending eligibility test.

 Most of the Recruiters find the suitable person based on the scores they have gotten in the eligibility.

**DISADVANTAGES**

 Person Job May get technical difficulty while taking the eligibility

 Job seeker may have trouble to contact recruiters directly.







**11 CONCLUSION**

The application has been developed to make job search easier .

The application that we have developed is user friendly .

User can find a job based on their skillset in the short period of time. The jobseeker certainly get benefit by using this application.

In the addition,Chatbot Has been implemented with the help of IBM whatson . The chatbot helps jobseeker and organization when they experience the difficulties**.**

**12 FUTURE SCOPE**

The linked in the wellknown application to find a job and stay connected with professional and organization.

The job seekers and organization use linked in to find a job.

In the future , There are lots of possibilities to enhance our project similar to linked in.

**13 APPENDIX** SOURCE CODE **Sprint 1: Index.html**

<!DOCTYPEhtml>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Log-in Page </title>

<link href="css/bootstrap.min.css" rel="stylesheet">

<script src="chat.js"></script>

</head>

<style>

.pad{

padding-top: 40%;

}

.bold{

font-weight: bolder;

}

.backgroundcolour{

background-color: beige;

}

.font{

font-weight: bolder; color:whitesmoke;

}

</style>

<body>

<div class="container">

<div class="row">

<div class="col-sm-6">

</div>

<div class="col-sm-6">

<div class="pad ">

<center><img src="images/logo.png" class="img-responsive" width="20%" height="20%"> </center> <HR>

<form>

<div

class="input-group">

<span

class="input-group-addon"><i class="glyphicon glyphicon-user"></i></span>

<input id="email" type="text" class="form-control" name="email" placeholder="Enter Your Registered Email ID" required>

class="input-group">

</div>

<br>

<div

<span

class="input-group-addon"><i class="glyphicon glyphicon-lock"></i></span>

<input id="password" type="password" class="form- control" name="password" placeholder="Enter Your Password" required>

</div>

<hr>

<div class="row">

<div class="col-sm-6">

<button type="submit" id="" class="btn btn- success btn-block"> <span class="font">LOG IN

</span></button>

</div>

<div class="col-sm-6">

<button id="" class="btn btn-danger btn- block"><a href="forget\_password.html"><span class="font">FORGET PASSWORD

</span></a></button>

<br>

</div>

</div>

</form>

<a href="signup.html"> <button type="submit" id="" class="btn btn-primary btn-block"><span class="font">CREATE AN ACCOUNT </span> </button>

</a>

</div>

</div>

</div>

</div>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="chat.js"></script>

</body>

</html>

**FORGET-PASSWORD.html**

<!DOCTYPEhtml>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Responsive Sample</title>

<link href="css/bootstrap.min.css" rel="stylesheet">

</head>

<style>

.pad{

padding-top: 20%;

padding-left: 20%;

padding-right: 20%;

}

.bold{

font-weight: bolder;

}

</style>

<body>

<div class="container">

<div class="pad">

<form>

<center> <H3 class="bold">FORGET PASSWORD</H3></center> <hr>

<div class="form-group">

<label for="email">Email

address:</label>

<input type="email"

class="form-control" id="email" placeholder="Enter Your Registered Email Addtress">

</div>

<div class="form-group">

<label

for="pwd">OTP</label>

<input type="password"

class="form-control" id="pwd" placeholder="Enter your OTP ">

</div>

<button type="submit" class="btn btn-default">Recover Password</button>

</form>

</div>

</div>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

SIGN UP.html

<!DOCTYPE

html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Creating New Account

</title>

<link href="css/bootstrap.min.css" rel="stylesheet">

</head>

<style>

.pad{

padding-top: 10%;

padding-bottom: 10%;

}

.bold{

font-weight: bolder;

}

</style>

<body>

<div class="container">

class="pad">

<div

<h3

class="bold">CREATE AN ACCOUNT</h3>

<HR>

<form> class="form-group">

<div

<label for="name">First Name</label>

<input type="name" class="form-control" id="name" placeholder="Enter Your First Name">

</div> class="form-group">

<label for="name">Last Name</label>

<div

<input type="name" class="form-control" id="name" placeholder="Enter Your Last Name">

</div>

<div class="form-group">

<label for="email">Email address:</label>

<input type="email" class="form-control" id="email" placeholder="Enter Your Email ID">

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" id="pwd" placeholder="Enter Your Password">

</div> class="form-group">

<div

<label for="repwd">Reenter Password:</label>

<input type="password" class="form-control" id="repwd" placeholder="Enter Your Password Again">

</div>

<button type="button" id="" class="btn btn-primary btn-block btn-danger">CREATE AN ACCOUNT</button>

<hr>

<button type="submit" id="login\_already" class="btn btn- primary btn-block btn-success">Do you have an Account Already

?</button>

</div>

</form>

</div>

</div>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="chat.js"></script>

</body>

</html>

**BACK END :**

import os import ibm\_db import bcrypt

#import send\_from\_directory

app = Flask( name ) #initialising flask app.secret\_key = ''

PEOPLE\_FOLDER = os.path.join('static', 'people\_photo') import ibm\_db

conn = ibm\_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f- 46df-8954-

7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;P ORT=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRoo tCA.crt;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45Raj9A0rr",'', '')

@app.route("/") #defining the routes for the home() funtion (Multiple routes can be used as seen here) @app.route("/home")

def home():

return render\_template("home.html") #rendering our home.html contained within /templates @app.route("/login")

def login():

if request.method == 'POST': email = request.form['email']

password = request.form['password'] if not email or not password:

return render\_template('login.html',error='Please fill all fields')

query = "SELECT \* FROM USER WHERE email=?" stmt = ibm\_db.prepare(conn, query) ibm\_db.bind\_param(stmt,1,email) ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt) print(isUser,password)

if not isUser: return

render\_template('login.html',error='Invalid Credentials')

isPasswordMatch = bcrypt.checkpw(password.encode('utf- 8'),isUser['PASSWORD'].encode('utf-8'))

if not isPasswordMatch: return

render\_template('login.html',error='Invalid Credentials') session['email'] = isUser['EMAIL']

return redirect(url\_for('user\_dashboard')) return render\_template('login.html',name='Home')

app.config["UPLOAD\_DIR"] = "uploads" @app.route("/register",methods=['GET','POST']) def register():

if request.method == 'POST': email = request.form['email']

password = request.form['password'] firstname=request.form['firstname'] middlename=request.form['middlename'] lastname=request.form['lastname'] dob=request.form['dob'] age=request.form['age'] drno=request.form['drno'] landmark1=request.form['lnmk1'] landmark2=request.form['lnmk2'] village =request.form['village'] district=request.form['district'] pincode=request.form['pincode'] country=request.form['country']

country\_code=request.form['countryCode']

phone=request.form['phone'] schlname10=request.form['schlname10'] maxmark10=request.form['maxmark10'] actmark10=request.form['actmark10'] schlname212=request.form['schlname212'] maxmark212=request.form['maxmark212'] actmark212=request.form['actmark212'] clgname=request.form['clgname'] clgmaxmark=request.form['clgmaxmark'] clgactmark=request.form['clgactmark'] stream=request.form['stream'] graduation=request.form['graduation'] grayear=request.form['grayear'] branch=request.form['branch'] skills=request.form['form[]'] q1=request.form['Q1'] q2=request.form['Q2']

q3=request.form['Q3'] q4=request.form['Q4'] q5=request.form['Q5'] position=request.form['position'] experience=request.form['expyears'] designation=request.form['designation'] tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD\_DIR'],tum.filename)) hash=bcrypt.hashpw(password.encode('utf-

8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"

stmt = ibm\_db.prepare(conn, query) ibm\_db.bind\_param(stmt,1,email) ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not is User:

insert\_sql = "INSERT INTO JOBSEEKER (EMAIL, PASSWORD, FIRSTNAME, MIDDLENAME, LASTNAME, DOB, AGE, DOOR\_NO, LANDMARK1, LANDMARK2,VILLAGE, DISTRICT, PINCODE, COUNTRY, COUNTRY\_CODE, PHONE, SCHOOL\_NAME, SC\_MAX\_MARK, SC\_ACT\_MARK,HSC\_SCHOOL\_NAME,HSC\_MAX\_MARK, HSC\_ACT\_MARK,COLLEGE\_NAME,COLLEGE\_MAX,COLLEGE\_ACT, STREAM, GRADUATION, GRADUATION\_YEAR, BRANCH, SKILLS, PRESENTATION, HACKATHON, CODING, WON, PRIZE\_WON, POSITION, EXPYEARS,DESIGNATION, RESUME) VALUES (?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,

?,?,?,?,?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql) ibm\_db.bind\_param(prep\_stmt, 1, email)

ibm\_db.bind\_param(prep\_stmt, 2, password)

ibm\_db.bind\_param(prep\_stmt, 3, firstname)

ibm\_db.bind\_param(prep\_stmt, 4, middlename)

ibm\_db.bind\_param(prep\_stmt, 5, lastname)

ibm\_db.bind\_param(prep\_stmt, 6, dob)

ibm\_db.bind\_param(prep\_stmt, 7, age)

ibm\_db.bind\_param(prep\_stmt, 8, drno)

ibm\_db.bind\_param(prep\_stmt, 9, landmark1)

ibm\_db.bind\_param(prep\_stmt, 10, landmark2)

ibm\_db.bind\_param(prep\_stmt, 11, village)

ibm\_db.bind\_param(prep\_stmt, 12, district)

ibm\_db.bind\_param(prep\_stmt, 13, pincode)

ibm\_db.bind\_param(prep\_stmt, 14, country)

ibm\_db.bind\_param(prep\_stmt, 15, country\_code)

ibm\_db.bind\_param(prep\_stmt, 16, phone)

ibm\_db.bind\_param(prep\_stmt, 17, schlname10)

ibm\_db.bind\_param(prep\_stmt, 18, maxmark10)

ibm\_db.bind\_param(prep\_stmt, 19, actmark10)

ibm\_db.bind\_param(prep\_stmt, 20, schlname212)

ibm\_db.bind\_param(prep\_stmt, 21, maxmark212)

ibm\_db.bind\_param(prep\_stmt, 22, actmark212)

ibm\_db.bind\_param(prep\_stmt, 23, clgname)

ibm\_db.bind\_param(prep\_stmt, 24, clgmaxmark)

ibm\_db.bind\_param(prep\_stmt, 25, clgactmark)

ibm\_db.bind\_param(prep\_stmt, 26, stream)

ibm\_db.bind\_param(prep\_stmt, 27, graduation)

ibm\_db.bind\_param(prep\_stmt, 28, grayear)

ibm\_db.bind\_param(prep\_stmt, 29, branch)

ibm\_db.bind\_param(prep\_stmt, 30, skills)

ibm\_db.bind\_param(prep\_stmt, 31, q1)

ibm\_db.bind\_param(prep\_stmt, 32, q2)

ibm\_db.bind\_param(prep\_stmt, 33, q3)

ibm\_db.bind\_param(prep\_stmt, 34, q4)

ibm\_db.bind\_param(prep\_stmt, 35, q5)

ibm\_db.bind\_param(prep\_stmt, 36, position)

ibm\_db.bind\_param(prep\_stmt, 37, experience)

ibm\_db.bind\_param(prep\_stmt, 38, designation)

ibm\_db.bind\_param(prep\_stmt, 39, tum) ibm\_db.execute(prep\_stmt)

return render\_template('register.html') else:

return render\_template('register.html',error='Invalid Credentials')

return render\_template('register.html',name='Home') @app.route("/orgregister")

def orgregister():

return render\_template("orgregister.html") #rendering our register.html contained within /templates @app.route("/user\_dashboard")

def user\_dashboard():

return render\_template("user\_dashboard.html") @app.route('/logout')

def logout():

session.pop('email', None) return redirect(url\_for('login'))

if \_\_name == " main ": #checking if name\_\_'s value is

' main '. name is an python environment variable who's value will always be ' main ' till this is the first

instatnce of app.py running

app.run(debug=True,port=8080) #running flask (Initalised on line 4)

SPRINT 2

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial- scale=1">

<title>Log-in Page </title>

<link href="css/bootstrap.min.css" rel="stylesheet">

<script src="chat.js"></script>

</head>

<style>

.pad{

padding-top: 40%;

}

.bold{

font-weight: bolder;

}

.backgroundcolour{ background-color: beige;

}

.font{

font-weight: bolder; color:whitesmoke;

}

</style>

<body>

<br>

<div class="container">

<div class="row">

<div class="col-sm-6">

<div class="pad ">

<center><img src="images/ORGANIZATION\_LOGO.png" class="img-responsive" width="20%" height="20%"> </center> <BR>

<form>

<div class="input-group">

<span class="input-group- addon"><i class="glyphicon glyphicon-user"></i></span>

<input id="email" type="text" class="form-control" name="email" placeholder="Enter Your Organization Registered Email ID" >

</div>

<br>

<div class="input-group">

<span class="input-group- addon"><i class="glyphicon glyphicon-lock"></i></span>

<input id="password" type="password" class="form-control" name="password" placeholder="Enter Your Password">

</div>

<hr>

<div class="row">

<div class="col-sm-6">

<button type="submit" id="" class="btn btn-success btn- block"> <span class="font">LOG IN </span></button>

</div>

<div class="col-sm-6">

<button id="" class="btn btn-danger btn-block">

<a href=""><span class="font">FORGET PASSWORD

</span></a></button>

</div>

</div> <br>

</form>

<ahref="Organization/Create\_Account/org\_create\_account.html">

<button type="submit" id="" class="btn btn-primary btn- block"><span class="font">CREATE AN ACCOUNT </span> </button>

</a>

</div>

</div>

<div class="col-sm-6">

<div class="pad ">

<center><img src="images/logo.png" class="img-responsive" width="20%" height="20%"> </center> <br>

group">

<form>

<div class="input-

<span class="input-

group-addon"><i class="glyphicon glyphicon-user"></i></span>

<input id="email" type="text" class="form-control" name="email" placeholder="Enter Your Registered Email ID" required>

</div>

<br>

group">

<div class="input-

<span class="input-

group-addon"><i class="glyphicon glyphicon-lock"></i></span>

<input id="password" type="password" class="form-control" name="password" placeholder="Enter Your Password" required>

</div>

<hr>

6">

<div class="row">

<div class="col-sm-

<button type="submit" id="" class="btn btn-success btn-block"> <span class="font">LOG IN </span></button>

6">

</div>

<div class="col-sm-

<button id="" class="btn btn-danger btn-block">

<a href="forget\_password.html">

<span class="font">FORGET PASSWORD </span></a></button>

</div>

</div> <br>

</form>

<a href="signup.html">

<button type="submit" id="" class="btn btn-primary btn-block">

<span class="font">CREATE AN ACCOUNT </span> </button>

</a></div>

</div>

</div>

</div>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="chat.js"></script>

</body>

</html**>**

FORGET-PASSWORD.html

<!DOCTYPEhtml>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Responsive Sample</title>

<link href="css/bootstrap.min.css" rel="stylesheet">

</head>

<style>

.pad{

padding-top: 20%;

padding-left: 20%;

padding-right: 20%;

}

.bold{

font-weight: bolder;

}

</style>

<body>

<div class="container">

<div class="pad">

<form>

<center> <H3 class="bold">FORGET PASSWORD</H3></center> <hr>

<div class="form-group">

<label for="email">Email

address:</label>

<input type="email"

class="form-control" id="email" placeholder="Enter Your Registered Email Addtress">

</div>

<div class="form-group">

<label

for="pwd">OTP</label>

<input type="password"

class="form-control" id="pwd" placeholder="Enter your OTP ">

</div>

<button type="submit" class="btn btn-default">Recover Password</button>

</form>

</div>

</div>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

</html>

**SIGNUP.html**

<!DOCTYPE

html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Creating New Account

</title>

<link href="css/bootstrap.min.css" rel="stylesheet">

</head>

<style>

.pad{

padding-top: 10%;

padding-bottom: 10%;

}

.bold{

font-weight: bolder;

}

</style>

<body>

<div class="container">

class="pad">

<div

<h3

class="bold">CREATE AN ACCOUNT</h3>

<HR>

<form> class="form-group">

<div

<label for="name">First Name</label>

<input type="name" class="form-control" id="name" placeholder="Enter Your First Name">

</div> class="form-group">

<div

<label for="name">Last Name</label>

<input type="name" class="form-control" id="name" placeholder="Enter Your Last Name">

</div>

<div class="form-group">

<label for="email">Email address:</label>

<input type="email" class="form-control" id="email" placeholder="Enter Your Email ID">

</div>

<div class="form-group">

<label for="pwd">Password:</label>

<input type="password" class="form-control" id="pwd" placeholder="Enter Your Password">

</div> class="form-group">

<div

<label for="repwd">Reenter

Password:</label>

<input type="password" class="form-control" id="repwd" placeholder="Enter Your Password Again">

</div>

<button type="button" id="" class="btn btn-primary btn-block btn-danger">CREATE AN ACCOUNT</button>

<hr>

<button type="submit" id="login\_already" class="btn btn- primary btn-block btn-success">Do you have an Account Already

?</button>

</div>

</form>

</div>

</div>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="chat.js"></script>

</body>

</html>

HOME PAGE :

Forgot password.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,

initial-scale=1">

<title>Responsive Sample</title>

<link href="css/bootstrap.min.css"

rel="stylesheet">

</head>

<style>

.pad{

padding-top: 20%;

padding-left: 20%;

padding-right: 20%;

}

.bold{

font-weight: bolder;

</style>

}

<body>

<div class="container">

<div class="pad">

<form>

<center> <H3 class="bold">FORGET

PASSWORD</H3></center> <hr>

<div class="form-group">

<label for="email">Email

address:</label>

<input type="email" class="formcontrol"

id="email" placeholder="Enter Your Registered Email

Addtress">

</div>

<div class="form-group">

<label for="pwd">OTP</label>

<input type="password" class="formcontrol"

id="pwd" placeholder="Enter your OTP ">

</div>

<button type="submit" class="btn btndefault">

Recover Password</button>

</form>

</div>

</div>

</html>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

</body>

Signup.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width,

initial-scale=1">

<title>Creating New Account </title>

<link href="css/bootstrap.min.css"

rel="stylesheet">

</head>

<style>

.pad{

padding-top: 10%;

padding-bottom: 10%;

}

.bold{

font-weight: bolder;

}

</style>

<body>

<div class="container">

ACCOUNT</h3>

<div class="pad">

<h3 class="bold">CREATE AN

<HR>

<form>

<div class="form-group">

<label

for="name">First Name</label>

<input type="name"

class="form-control" id="name" placeholder="Enter Your First

Name">

</div>

<div class="form-group">

<label

for="name">Last Name</label>

<input type="name"

class="form-control" id="name" placeholder="Enter Your Last

Name">

</div>

<div class="form-group">

for="email">Email address:</label>

<label

<input type="email"

class="form-control" id="email" placeholder="Enter Your Email

ID">

for="pwd">Password:</label>

</div>

<div class="form-group">

<label

<input

type="password" class="form-control" id="pwd" placeholder="Enter

Your Password">

</div>

<div class="form-group">

<label

for="repwd">Reenter Password:</label>

<input

type="password" class="form-control" id="repwd"

placeholder="Enter Your Password Again">

</div>

<button

type="button" id="" class="btn btn-primary btn-block btndanger">

CREATE AN ACCOUNT</button>

<hr>

<button

type="submit" id="login\_already" class="btn btn-primary btnblock

btn-success">Do you have an Account Already ?</button>

</div>

</form>

</div>

</div>

</html>

<script src="js/jquery.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="chat.js"></script>

</body>

BACK END APP.PY

from flask import \*#importing flask (Install it using python -m

pip install flask)

import os

import ibm\_db

import bcrypt

#import send\_from\_directory

app = Flask( name ) #initialising flask

app.secret\_key = ''

PEOPLE\_FOLDER = os.path.join('static', 'people\_photo')

import ibm\_dbconn =

ibm\_db.connect("DATABASE=bludb;HOSTNAME=54a2f15b-5c0f-46df-8954-

7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT

=32733;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.cr

t;PROTOCOL=TCPIP;UID=phb43134;PWD=mTTnIE45Raj9A0rr",'','')

@app.route("/") #defining the routes for the home() funtion

(Multiple routes can be used as seen here)

@app.route("/home")

def home():

return render\_template("index/INDEX.html") #rendering our

home.html contained within /templates

@app.route("/login")

def login():

if request.method == 'POST':

email = request.form['email']

password = request.form['password']

if not email or not password:

return render\_template('login.html',error='Please

fill all fields')

query = "SELECT \* FROM USER WHERE email=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

print(isUser,password)

if not isUser:

return render\_template('login.html',error='Invalid

Credentials')

isPasswordMatch = bcrypt.checkpw(password.encode('utf-

8'),isUser['PASSWORD'].encode('utf-8'))

if not isPasswordMatch:

return render\_template('login.html',error='Invalid

Credentials')

session['email'] = isUser['EMAIL']

return redirect(url\_for('user\_dashboard'))

return render\_template('login.html',name='Home')

app.config["UPLOAD\_DIR"] = "uploads"

@app.route("/register",methods=['GET','POST'])

def register():

if request.method == 'POST':

email = request.form['email']

password = request.form['password']

firstname=request.form['firstname']

middlename=request.form['middlename']

lastname=request.form['lastname']

dob=request.form['dob']

age=request.form['age']

drno=request.form['drno']

landmark1=request.form['lnmk1']

landmark2=request.form['lnmk2']

village =request.form['village']

district=request.form['district']

pincode=request.form['pincode']

country=request.form['country']

country\_code=request.form['countryCode']

phone=request.form['phone']

schlname10=request.form['schlname10']

maxmark10=request.form['maxmark10']

actmark10=request.form['actmark10']

schlname212=request.form['schlname212']

maxmark212=request.form['maxmark212']

actmark212=request.form['actmark212']

clgname=request.form['clgname']

clgmaxmark=request.form['clgmaxmark']

clgactmark=request.form['clgactmark']

stream=request.form['stream']

graduation=request.form['graduation']

grayear=request.form['grayear']

branch=request.form['branch']

skills=request.form['form[]']

q1=request.form['Q1']

q2=request.form['Q2']

q3=request.form['Q3']

q4=request.form['Q4']

q5=request.form['Q5']

position=request.form['position']

experience=request.form['expyears']

designation=request.form['designation']

tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD\_DIR'],tum.filename))

hash=bcrypt.hashpw(password.encode('utf-

8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM ORGANIZATION WHERE EMAIL=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO ORGANIZATION (ORGANIZATION\_NAME,

ORGANIZATION\_ID, ORGANIZATION\_ADDRESS,SET\_PASSWORD,

REENTERPASSWORD) VALUES (?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, ORGANIZATION\_NAME)

ibm\_db.bind\_param(prep\_stmt, 2, ORGANIZATION\_ID)

ibm\_db.bind\_param(prep\_stmt, 3, ORGANIZATION\_ADDRESS)

ibm\_db.bind\_param(prep\_stmt, 4, SET\_PASSWORD)

ibm\_db.bind\_param(prep\_stmt, 5, REENTERPASSWORD)

ibm\_db.execute(prep\_stmt)

return

render\_template('Organization\Create\_Account\org\_create\_account.

html')

else:

return

render\_template('Organization\Create\_Account\org\_create\_account.

html',error='Invalid Credentials')

return render\_template('create\_account.html',name='Home')

@app.route("/orgregister")

def orgregister():

return render\_template("org\_create\_account.html") #rendering

our register.html contained within /templates

@app.route("/user\_dashboard")

def user\_dashboard():

return render\_template("user\_dashboard.html")

@app.route('/logout')

def logout():

session.pop('email', None)

return redirect(url\_for('login'))

if name == "\_\_main ": #checking if name\_\_'s value is

' main '. name is an python environment variable who's

value will always be ' main ' till this is the first instatnce

of app.py running

app.run(debug=True,port=8080) #running flask (Initalised on

line 4)

hash=bcrypt.hashpw(password.encode('utf-

8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO JOBSEEKER (FIRSTNAME,

LASTNAME,EMAIL\_ADDRESSS,PWD,REPWD=?"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, firstname)

ibm\_db.bind\_param(prep\_stmt, 2, lastname)

ibm\_db.bind\_param(prep\_stmt, 3, email\_address)

ibm\_db.bind\_param(prep\_stmt, 4, pwd)

ibm\_db.bind\_param(prep\_stmt, 5, repwd)

ibm\_db.execute(prep\_stmt)

return render\_template('Job Seeker\signup.html')

else:

return render\_template('Job

Seeker\signup.html',error='Invalid Credentials')

return render\_template('signup.html',name='Home')

JAVASCRIPT:

window.watsonAssistantCh

atOptions= {

integrationID: "d73273d3-3f44-

4304-84ee-8fd243016d1d", // The ID of

this integration.

region: "jp-tok", // The region

your integration is hosted in.

serviceInstanceID: "81229104-ee6b-

46da-ac1c-67ede110663a", // The ID of

your service instance.

onLoad: function(instance) {

instance.render(); }

};

setTimeout(function(){

const

t=document.createElement('script');

t.src="https://webchat.

global.assistant.watson.appdomain

.cloud/versions/" +

(window.watsonAssistantChatOptions.cli

entVersion || 'latest') +

"/WatsonAssistantChatEntry.js";

document.head.appendChild(t);

});

**Verify.py**

from flask import request

from flask\_restful import Resource

from utils.tokener import confirm\_token

from utils.dbQuery import insertQuery

class Verify(Resource):

data=request.json

token=data["token"]

email=confirm\_token(token)

if(not email):

return {"status":"Couldn't verify"},400

res=insertQuery("UPDATE user set verified=True where

email=?",(email,))

if(not res):

return {"status":"Couldn't Update"},400

return {"status":"verified successfully"},200

**SPRINT :3**

**FRONT END**

DASHBROAD.html

<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initialscale=

1">

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst

rap.min.css">

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m

in.js"></script>

<script

src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra

p.min.js"></script>

<style>

.rus{

border-radius: 0%

};

.black{

color:black;

}

</style>

</head>

<body>

<!

========================================================-->

<nav class="navbar navbar-inverse rus">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Skill and Job

Recommender</a>

</div>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-login"></

span> Log out</a></li>

</ul>

</div>

</nav>

<div class="container">

<ul class="nav nav-tabs">

<li class="active"><a data-toggle="tab"

href="#home">Profile</a></li>

<li><a data-toggle="tab" href="#menu1">Eligibility

Test</a></li>

<li><a data-toggle="tab" href="#menu2">Find Job</a></li>

</ul>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<h3>PROFILE</h3>

<hr>

<form>

<div class="alert alert-danger">

<div class="form-group">

<label for="name"><span clss="black">Name:

</span></label>

<input type="text" class="form-control" id="name"

value="Mahadevan">

</div>

<div class="form-group">

<label for="email"><span class="black">Email Address

</span></label>

<input type="text" class="form-control" id="email"

value="Mahadevan@gmaiil.com">

</div>

<div class="form-group">

<label for="email"><span class="black">Mobile No

</span></label>

<input type="text" class="form-control" id="email"

value="9003824096">

</div>

<div class="form-group">

<label for="quali"><span class="black">What is your

Nationality </span></label>

<input type="text" class="form-control" id="quali"

value="India">

</div>

<div class="form-group">

<label for="quali">What is your

Qualification</label>

<input type="text" class="form-control" id="quali"

value="BE">

</div>

<div class="form-group">

<label for="quali">What year did you complete your

bachelor Degree?</label>

<input type="text" class="form-control" id="quali"

value="2023">

</div>

<div class="form-group">

<label for="experience">How many years of work

experience do you have ?</label>

<input type="text" class="form-control"

id="experience" value="0">

</div>

<div class="form-group">

<label for="place">Where have you graduated high

school degree ?</label>

<input type="text" class="form-control" id="place"

value="Government high school Ariyalur">

</div>

:</label>

<div class="form-group">

<label for="SSLC">What is your SSLC Percentage

<input type="text" class="form-control" id="SSLC"

value="90">

</div>

:</label>

<div class="form-group">

<label for="HSC">What is your HSC Percentage

<input type="text" class="form-control" id="HSC"

value="90">

</div>

<div class="form-group">

<label for="degree">What is your Percentage in

bachelor Degree:</label>

<input type="text" class="form-control" id="degree"

value="90.5">

</div>

<textarea class="form-control" id="aboutyou" rows="3"

placeholder="Write few words about yourself"></textarea> <br>

<br>

<button type="submit" class="btn btn-primary btnsuccess">

Save changes</button> <br> <br>

</div>

</form>

</div>

<div id="menu1" class="tab-pane fade">

<h3>Eligibility Test </h3> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the

screen?</label>

<div class="radio">

<label><input type="radio"

name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the

screen?</label>

<div class="radio">

<label><input type="radio"

name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the

screen?</label>

<div class="radio">

<label><input type="radio"

name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the

screen?</label>

<div class="radio">

<label><input type="radio"

name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="switch">switch</label>

</div>

</div> <hr>

<div class="alert alert-info">

<label>What function is used to display output on the

screen?</label>

<div class="radio">

<label><input type="radio"

name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="switch">switch</label>

</div>

</div><hr>

<div class="alert alert-info">

<label>What function is used to display output on the

screen?</label>

<div class="radio">

<label><input type="radio"

name="print">printf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="scanf">scanf()</label>

</div>

<div class="radio">

<label><input type="radio"

name="switch">switch</label>

</div>

</div>

<button type="submit" class="btn btn-primary

btn-success">Submit the test</button> <br> <br>

</div>

<div id="menu2" class="tab-pane fade">

<h3>Find the job</h3> <hr>

<br>

<div class="container">

<form>

<div class="input-group">

<input type="text" class="form-control"

id="myInput" placeholder="Look for a job">

<div class="input-group-btn">

type="submit">

search"></i>

<button class="btn btn-default"

<i class="glyphicon glyphicon-

</button>

</div>

</div>

</form> <br> <br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Java

Developer</h4>

<h5 class="media-heading"> Company Name : JK Software

Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btnsuccess">

Apply </button> <br> <hr>

</div>

</div>

</div>

<br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">.NET

Developer</h4>

<h5 class="media-heading"> Company Name : JK Software

Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btnsuccess">

Apply </button> <br> <hr>

</div>

</div>

</div>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">PHP

Developer</h4>

<h5 class="media-heading"> Company Name : JK Software

Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btnsuccess">

Apply </button> <br> <hr>

</div>

</div>

</div>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">UX /UI

Designer</h4>

<h5 class="media-heading"> Company Name : JK Software

Solution</h4>

<p>Job Location : Chennai</p>

<p> Salary : 45000</p>

<button type="submit" class="btn btn-primary btnsuccess">

Apply </button> <br> <hr>

</div>

</div>

</div>

</div>

</div>

</div>

<script>

$(document).ready(function(){

$("#myInput").on("keyup", function() {

var value = $(this).val().toLowerCase();

$("#myDIV \*").filter(function() {

$(this).toggle($(this).text().toLowerCase().indexOf(value)

> -1)

});

});

});

</script>

</body>

</html>

BACK END :

from flask import \*#importing flask (Install it using python -m

pip install flask)

import os

import ibm\_db

import bcrypt

#import send\_from\_directory

app = Flask( name ) #initialising flask

app.secret\_key = ''

PEOPLE\_FOLDER = os.path.join('static', 'people\_photo')

import ibm\_db

conn = ibm\_db.connect("DATABASE=bludb;

HOSTNAME=54a2f15b-5c0f-46df-8954-

7e38e612c2bd.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT

=32733;SECURITY=SSL;

SSLServerCertificate=DigiCertGlobalRootCA.crt;PROTOCOL=TCPIP;UID

=phb43134;PWD=mTTnIE45Raj9A0rr",'','')

@app.route("/") #defining the routes for the home() funtion

(Multiple routes can be used as seen here)

@app.route("/home")

def dashboard():

return render\_template("dashboard.html")

def createjobseeker\_profile():

if request.method == 'POST':

name = request.form['name']

email = request.form['email']

mobileno=request.form['mobileno']

nationality=request.form['nationality']

qualification=request.form['qualification']

passout=request.form['passout']

experience=request.form['experience"];

degree=request.form['degree'];

per\_sslc=request.form['per\_sslc'];

per\_hsc=request.form['per\_hsc'];

per\_degree=request.form['per\_degree'];

tum = request.files['resume']

tum.save(os.path.join(app.config['UPLOAD\_DIR'],tum.filename))

hash=bcrypt.hashpw(password.encode('utf-

8'),bcrypt.gensalt())

query = "SELECT EMAIL FROM JOBSEEKER WHERE EMAIL=?"

stmt = ibm\_db.prepare(conn, query)

ibm\_db.bind\_param(stmt,1,email)

ibm\_db.execute(stmt)

isUser = ibm\_db.fetch\_assoc(stmt)

if not isUser:

insert\_sql = "INSERT INTO JOBSEEKER (name, email,

mobileno, nationality,

qualification, passout, experience,

degree,per\_sslc,per\_hsc,per\_degree) VALUES

(?,?,?,?,?,?,?,?,?,?,?,?)"

prep\_stmt = ibm\_db.prepare(conn, insert\_sql)

ibm\_db.bind\_param(prep\_stmt, 1, name)

ibm\_db.bind\_param(prep\_stmt, 2, email)

ibm\_db.bind\_param(prep\_stmt, 3, mobileno)

ibm\_db.bind\_param(prep\_stmt, 4, nationality)

ibm\_db.bind\_param(prep\_stmt, 5, qualification)

ibm\_db.bind\_param(prep\_stmt, 6, passout)

ibm\_db.bind\_param(prep\_stmt, 7, experience)

ibm\_db.bind\_param(prep\_stmt, 8, degree)

ibm\_db.bind\_param(prep\_stmt, 9, per\_sslc)

ibm\_db.bind\_param(prep\_stmt, 10, per\_hsc)

ibm\_db.bind\_param(prep\_stmt, 11, per\_degree)

ibm\_db.execute(prep\_stmt)

return render\_template('dashboard.html')

return render\_template('dashboard.html',name='Home')

@app.route('/logout')

def logout():

session.pop('email', None)

return redirect(url\_for('login'))

**SPRINT:4**

<!DOCTYPE html>

<html lang="en">

<head>

<title>Bootstrap Example</title>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initialscale=

1">

<link rel="stylesheet"

href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootst

rap.min.css">

<script

src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.m

in.js"></script>

<script

src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstra

p.min.js"></script>

<style>

.rus{

border-radius: 0%

};

.black{

color:black;

}

</style>

</head>

<body>

<!

========================================================-->

<nav class="navbar navbar-inverse rus">

<div class="container-fluid">

<div class="navbar-header">

<a class="navbar-brand" href="#">Skill and Job

Recommender</a>

</div>

<ul class="nav navbar-nav navbar-right">

<li><a href="#"><span class="glyphicon glyphicon-login"></

span> Log out</a></li>

</ul>

</div>

</nav>

<!

========================================================-->

<div class="container">

<ul class="nav nav-tabs">

<li class="active"><a data-toggle="tab" href="#home">Company

Profile</a></li>

<li><a data-toggle="tab" href="#menu2">Find

candidate</a></li>

</ul>

<div class="tab-content">

<div id="home" class="tab-pane fade in active">

<h3>COMPANY PROFILE</h3>

<hr>

<form>

<div class="alert alert-danger">

<div class="form-group">

<label for="name"><span clss="black">Company Name:

</span></label>

<input type="text" class="form-control" id="name"

value="Mahadevan">

</div>

<div class="form-group">

<label for="email"><span class="black">Company Email

Address </span></label>

<input type="text" class="form-control" id="email"

value="Mahadevan@gmaiil.com">

</div>

<div class="form-group">

<label for="email"><span class="black">Mobile No

</span></label>

<input type="text" class="form-control" id="email"

value="9003824096">

</div>

<div class="form-group">

<label for="quali"><span

class="black">Location</span></label>

<input type="text" class="form-control" id="quali"

value="India">

</div>

<textarea class="form-control" id="aboutyou" rows="3"

placeholder="Write few words about company"></textarea> <br>

<br>

<button type="submit" class="btn btn-primary btnsuccess">

Save changes</button> <br> <br>

</div>

</form>

</div>

<div id="menu2" class="tab-pane fade">

<h3>Find the Condidate</h3> <hr>

<br>

<div class="container">

<!

>

<form>

<div class="input-group">

<input type="text" class="form-control"

id="myInput" placeholder="Look for a candidate">

<div class="input-group-btn">

<button class="btn btn-default"

type="submit">

search"></i>

<i class="glyphicon glyphicon-

</button>

</div>

</div>

</form> <br> <br>

<! >

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Name :

Mahadevan</h4>

<h5 class="media-heading"> Qualification : B.E

(Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btnsuccess">

Know More </button> <br> <hr>

</div>

</div>

</div> <br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Name :

Manikandan</h4>

<h5 class="media-heading"> Qualification : B.E

(Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btnsuccess">

Know More </button> <br> <hr>

</div>

</div>

</div> <br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

</div>

<div class="media-body">

<h4 class="media-heading">Name :

Kavi</h4>

<h5 class="media-heading"> Qualification : B.E

(Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btnsuccess">

Know More </button> <br> <hr>

</div>

</div>

</div> <br>

<div style="width: 600px;">

<!-- Left-aligned -->

<div class="media">

<div class="media-left">

<img src="images/img\_avatar1.png"

class="media-object" style="width:60px">

Ram</h4>

</div>

<div class="media-body">

<h4 class="media-heading">Name :

<h5 class="media-heading"> Qualification : B.E

(Computer Science Engineering )</h4>

<button type="submit" class="btn btn-primary btnsuccess">

Know More </button> <br> <hr>

</div>

</div>

</div> <br>

</div>

</div>

</div>

<script>

$(document).ready(function(){

$("#myInput").on("keyup", function() {

var value = $(this).val().toLowerCase();

$("#myDIV \*").filter(function() {

$(this).toggle($(this).text().toLowerCase().indexOf(value) > -1)

});

});

});

</script>

</body>

</html>

**GITHUB & PROJECT DEMO LINK:**

All the tasks of developing the application

were uploaded on the github.

The github has been uploaded below.

https://github.com/IBM-EPBL/IBM-Project-25197-1659954891