

NAALAIYA THIRAN PROJECT – 2022
19ECI01-PROFESSIONAL READINESS FOR
INNOVATION, EMPLOYABILITY AND
ENTREPRENEURSHIP

SKILS AND JOB RECOMMANDER APPLICATION

A PROJECT REPORT

Submitted by

SAKTHIVEL S	1913078
SANJAY KUMAR E	1913082
SWATHI K	1913096
SATHISH P	1913505

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

KSR COLLEGE OF ENGINEERING, TIRUCHENGODE (637215)

(AUTONOMOUS)

ANNA UNIVERSITY: CHENNAI 600025

NOVEMBER 2022

KSR COLLEGE OF ENGINEERING
(AUTONOMOUS)
TIRUCHENGODE-637215
ANNA UNIVERSITY: CHENNAI 600 025
BONAFIDE CERTIFICATE

Certified that this report “**SKILL/JOB RECOMMENDER APPLICATION**” is the bonafide work of **SAKTHIVEL S** (1913078), **SANJAY KUMAR E** (1913082), **SWATHI K** (1913096) and **SATHISH P** (1913505) who carried out 19ECI01 Professional Readiness for Innovation, Employability and Entrepreneurship project offered by IBM and Anna University, Chennai.

SIGNATURE

Dr .N.S.NITHYA

FACULTY MENTOR

Associate Professor

Department of Computer

Science and Engineering

SIGNATURE

Dr. V. BABY ANITHA

FACULTY EVALUATOR

Associate Professor

Department of Computer

Science and Engineering

SIGNATURE

Dr. P.SENTHILKUMAR

PRINCIPAL

Professor

PROJECT CALENDER

Phase	Phase Description	Week	Dates	Activity Details
1	Preparation Phase (Pre-requisites, Registrations, Environment Set-up, etc.)	2	22 - 27 Aug 2022	Creation GitHub account & collaborate with Project repository in project workspace
2	Ideation Phase (Literature Survey, Empathize, Defining Problem Statement, Ideation)	2	29 Aug – 3rd Sept 2022	Literature survey (Aim, objective, problem statement and need for the project)
		3	5 - 10th Sept 2022	Preparing Empathy Map Canvas to capture the user Pains & Gains
		4	12 - 17 Sept 2022	Listing of the ideas using brainstorming session
3	Project Design Phase -I (Proposed Solution, Problem- Solution Fit, Solution Architecture)	5	19 - 24 Sept 2022	Preparing the proposed solution document
		6	26 Sept - 01 Oct 2022	Preparing problem - solution fit document &Solution Architecture
4	Project Design Phase -II (Requirement Analysis, Customer Journey, Data Flow Diagrams, Technology Architecture)	7	3 - 8 Oct 2022	Preparing the customer journey maps
		8	10 - 15 Oct 2022	Preparing the Functional Requirement Document & Data- Flow Diagrams and Technology Architecture
5	Project Planning Phase (Milestones & Tasks, Sprint Schedules)	9	17 - 22 Oct 2022	Preparing Milestone & Activity List, Sprint Delivery Plan
6	Project Development Phase (Coding & Solutioning, acceptance Testing, Performance Testing)	10	24 - 29 Oct 2022	Preparing Project Development Delivery of Sprint-1
		11	31 Oct - 5 Nov 2022	Preparing Project Development - Delivery ofSprint-2
		12	7 - 12 Nov 2022	Preparing Project Development - Delivery ofSprint-3
		13	14 - 19 Nov 2022	Preparing Project Development Delivery of Sprint-4

TABLE OF CONTENTS

CHAPTER NO	CONTENTS	PAGE NO
	LIST OF FIGURES	
	LIST OF TABLES	
1	INTRODUCTION 1.1 PROJECT OVERVIEW 1.2 PURPOSE	5
2	LITERATURE SURVEY 2.1 EXISTING PROBLEM 2.2 REFERENCE 2.3 PROBLEM STATEMENT	6
3	IDEATION & PROPOSED SOLUTION 3.1 EMPATHY MAP CANVAS 3.2 IDEATION AND BRAINSTORMING 3.3 PROPOSED SOLUTION 3.4 PROBLEM SOLUTION FIT 3.5 SOLUTION ARCHITECTURE	9
4	REQUIREMENT ANALYSIS 4.1 FUNCTIONAL REQUIREMENTS 4.2 NON FUNCTIONAL REQUIREMENTS	14
5	PROJECT DESIGN 5.1 DATA FLOW DIAGRAMS 5.2 SOLUTION AND TECHNICAL ARCHITECTURE 5.3 USER STORY	15
6	PROJECT PLANNING & SCHEDULING 6.1 SPRINT PLANNING AND ESTIMATION 6.2 SPRINT DELIVERY SCHEDULE	18
7	CODING & SOLUTIONING 7.1 CHATBOT INTEGRATION CODE 7.2 FLASK FRAMEWORK 7.3 DATA BASE SCHEMA	20
8	TESTING 8.1 TEST CASES 8.2 USER ACCEPTANCE TESTING	25
9	RESULTS 9.1 PERFORMANCE METRICS	26
10	ADVANTAGES & DISADVANTAGES 10.1 ADVANTAGES 10.2 DISADVANTAGES	27
11	CONCLUSION	27
12	FUTURE SCOPE	27
13	APPENDIX	28

1. INTRODUCTION

1.1 Project Overview:

To provide the right and plenty of opportunities to the fresher's and experienced candidates. We have come up with a skill recommender solution through which the fresher or the skilled person can register, log in, and find jobs with our help according to their ability, skill level, and knowledge. In this process, we are going to help them in both a human-centered way and by using technology. Further will be discussed below.

1.2 Purpose:

To provide equal and appropriate job/skill recommendations to all the people with their ability, skill level, and knowledge consideration. We are not going to filter the candidates based on our machine-analyzed data that the candidate entered in the resume or wherever. Because we found that many people said they are not getting the right recommendations for jobs and skill enhancement courses. So, we are approaching this problem in a more human-centric way to get the data, processing, and recommend skills and jobs. We are merging both technology and human in the right proportion to solve the real human problem.

2. LITERATURE SURVEY

2.1 Existing Problem:

How might we recommend jobs for people's specific skills or interests and vice-versa? In the job searching world, Job description's explanation, advertising, and selection criteria aren't the same during college and after college. It is entirely different. People are facing many practical issues when they searching jobs after their college days. As we are going to provide a different kind of job description alone will create a positive hopeful impact for fresher as well as experienced candidates. By this, we can satisfy both the fresher and experienced candidates who are all searching for jobs. As we are going to find the skills of the people through both the direct and indirect approach, we are able to get and find the right skill level for the people. In addition, we can get more insights about them. This creates a more positive impact and customer satisfaction in using our service and this is the existing problem to solve.

2.2 References:

S.NO	TITLE	AUTHOR & YEAR	DESCRIPTION	OUR LEARNINGS
1.	Job Recommendation System Using Profile Matching and Web Crawling	Deepali V Musale, Mamta K Nagpure, Kaumudini S Patil, Rukhsar F Sayyed May 2016.	The first type of recommendation is done through web portal by using keyword based search and second type of recommendation is done through profile matching and sending notification to the students. Thus proper job recommendations are provided to the students.	1. They use the more logical way approach. 2. Getting completely detailed information - Compare profiles - Recommendation.
2.	Enhanced Job Recommendation System.	Shivraj Hulbatte, Amit Wabale, Suraj Patil, Nikhilkumar Sathe Oct 2018	On the basis of this study, research and after implementation of algorithms, the collaborative filtering based algorithm is considered for its better performance and overall factors. Of course a lot of improvement and hybrid algorithms need to be implemented alongside collaborative filtering algorithm.	1. Started with similar approach that we have now for recommendation, but ended up with the statements of default sayings.

3.	Job Recommendation based on Job Seeker Skills: An Empirical Study.	Jorge Valverde-Rebaza, Ricardo Puma, Paul Bustios, Nathalia C. Silva March 2018.	In this paper, 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.	1. Collect the profiles & job offers through sites and compared with the relatively large number of keywords through certain text processing methods.
4.	A survey of job recommender systems.	Shaha T. Al-Otaibi ^{1*} , Mourad Ykhlef ² Aug 2012	This survey shows that several approaches for job recommendation have been proposed, and many techniques combined in order to produce the best fit between jobs and candidates.	1. Got an general overview about job recommendation systems and realized lot to learn. 2. Got new concepts or ideas for our project. Have to explore it.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas:



3.2 Ideation & Brainstorming:






3.3 Proposed Solution:





S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	How might we recommend jobs for people's specific skills or interests and vice-versa?
2.	Idea / Solution description	Specific public campaigns for grouping people based on their skills, Asking recruiters not to put detailed descriptions and asking them to put the skills they want. Based on user requests, show the job vacancies that mostly match with user skills.
3.	Novelty / Uniqueness	We are not going to filter the candidates based on our machine-analyzed data that the candidate entered in the resume or wherever. Because we found that many people said they are not getting the right recommendations for jobs and skill enhancement courses. So, we are approaching this problem in a more human-centric way to get the data, processing, and recommend skills and jobs. We are merging both technology and human in the right proportion to solve the real problem.

4.	Social Impact / Customer Satisfaction	<p>Job description explanation, advertising, and selection criteria aren't the same during college and after college. It is entirely different. People are facing many practical issues when they searching jobs after their college days. As we are going to provide a different kind of job description alone will create a positive hopeful impact for fresher as well as for experienced candidates. By this, we can satisfy both the fresher and experienced candidates who are all searching for jobs. As we are going to find the skills of the people through both the direct and indirect approach, we are able to get and find the right skill level for the people. In addition, we can get more insights about them. This creates a more positive impact and customer satisfaction in using our service.</p>
5.	Business Model (Revenue Model)	<ol style="list-style-type: none"> 1. Advertisement commissions by allowing skill development companies and job-providing agencies to promote their business to reach their potential users. 2. Premium services like mentor support, exclusive courses, etc...
6.	Scalability of the Solution	<p>We found the core problem among job seekers. Now, we are trying to solve it by simply finding users' skill levels in a more human-centered way and recommending jobs based on their category at all levels. We believed that this idea can be scalable. Because once we achieve/find an implementation approach to our above-mentioned primary task then,</p> <ol style="list-style-type: none"> 1. It can be expandable by providing offline support along with online. 2. End-to-end support for both skills identification, enhancement, job support, and more.

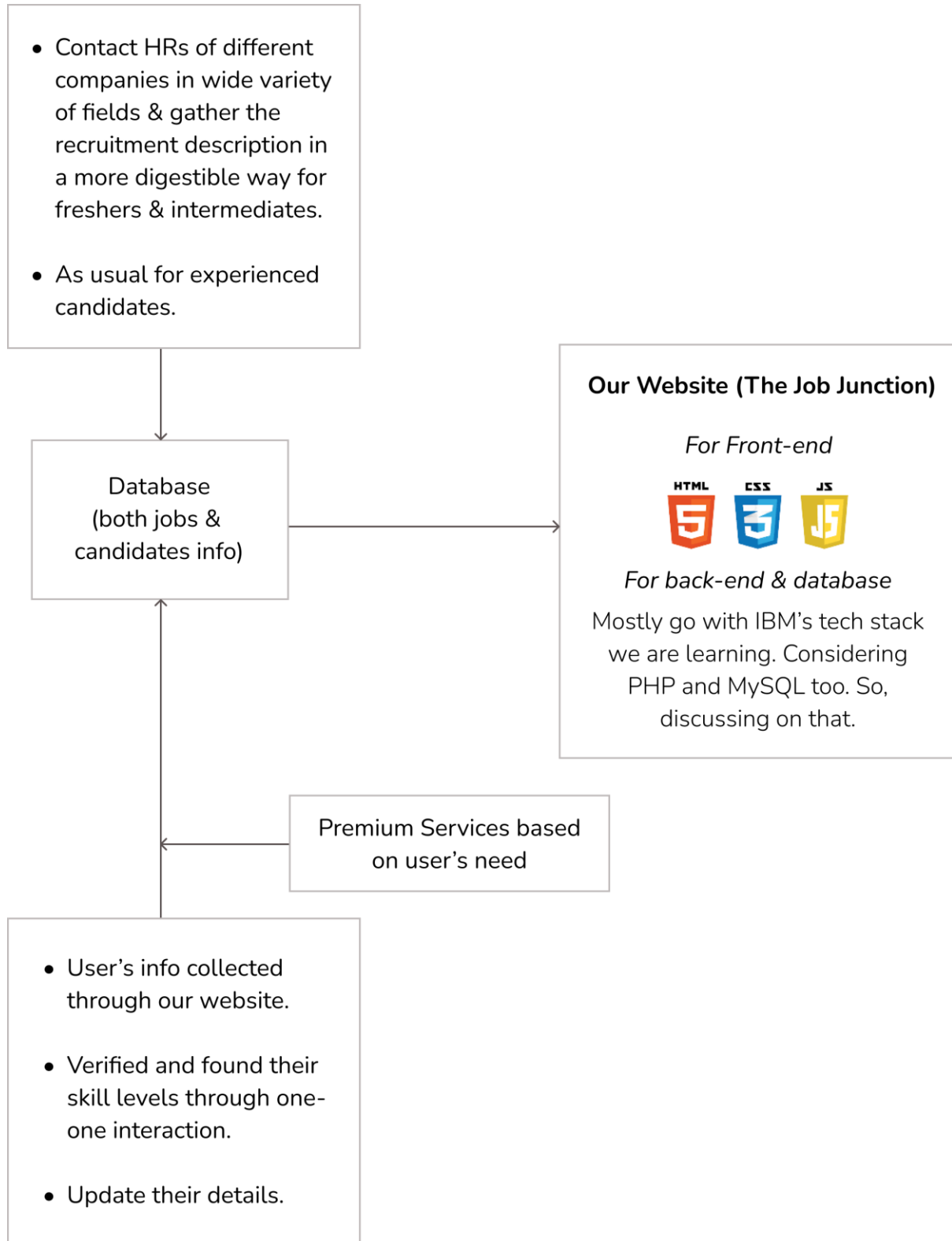
3.4 Problem Solution Fit:

<p>Define CS, fit into CC</p> <p>1. CUSTOMER SEGMENT(S)  College students who are in the stage of applying for jobs. Includes who all are found their interests and not found yet.</p>	<p>6. CUSTOMER CONSTRAINTS  Social media distraction, Language barrier and Lack of awareness.</p>	<p>5. AVAILABLE SOLUTIONS </p> <p>1. Job Seeking & Hiring Platforms: Most popular among professionals. It recommends jobs based on our requirement but not to the fullest. People received their domain based recommendation but it is not fit to their skill levels. Descriptions make them scary.</p> <p>2. College Placements: People simply attend placement drives without even knowing the detailed description of a job, roles and responsibilities. Aptitude - Coding - Technical interview - HR round. But people outside college (fresher) struggle a lot with the above platform's job description. It occurs generally in the field irrespective of these platforms too.</p> <p>Explore AS, differentiate</p>
---	---	---

<p>2. JOBS-TO-BE-DONE / PROBLEMS </p> <p>1. Not getting proper job recommendations based on their skill level or proficiency in that particular field.</p> <p>2. Not find their interests yet to move ahead in their career.</p> <p>3. Not proper guidance and support for them to lead them in their right path.</p> <p>Focus on J&P, fit into BE, understand</p>	<p>9. PROBLEM ROOT CAUSE </p> <p>1. Job Recommendations aren't based on the people's need fully. It just give outputs for the input. It has to be based deeply with the people's mindset and their skill levels.</p> <p>2. Job description and placement methodologies aren't same inside and outside the college for fresher. For more (refer #5).</p>	<p>7. BEHAVIOUR </p> <p>1. Contact their relatives, seniors and friends who works in their interested fields to know about the opportunities, to clear their doubts and for guidance.</p> <p>2. Googling about the jobs.</p> <p>3. Trial and error method.</p>
--	--	---

<p>3. TRIGGERS </p> <p>1. College seniors / College's recommendation.</p> <p>2. Close friends sharing the product.</p> <p>3. Product in their mother language.</p> <p>4. Low cost.</p> <p>5. Testimonials, Reviews, and Achievements.</p> <p>4. EMOTIONS: </p> <p>BEFORE</p> <p>1. Self-doubt.</p> <p>2. Low self-esteem.</p> <p>AFTER</p> <p>1. Self analysis (no self-doubt).</p> <p>2. Skill found (high self-esteem).</p> <p>3. Job recommendation based on their skill level (happy).</p>	<p>10. YOUR SOLUTION </p> <p>We found the core problem among job seekers. Now, we are trying to solve it by simply finding users' skill levels in a more human-centred way and recommending jobs based on their category at all levels. We believed that this idea can be scalable. Because once we achieve/find an implementation approach to our above-mentioned primary task then,</p> <p>1. It can be expandable by providing offline support along with online.</p> <p>End-to-end support for both skills identification, enhancement, job support, and more.</p>	<p>8. CHANNELS of BEHAVIOUR </p> <p>ONLINE</p> <p>1. Analyzing Job searching platforms.</p> <p>2. Googling.</p> <p>OFFLINE</p> <p>1. Contact peoples.</p>
--	--	--

3.5 Solution Architecture:



4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

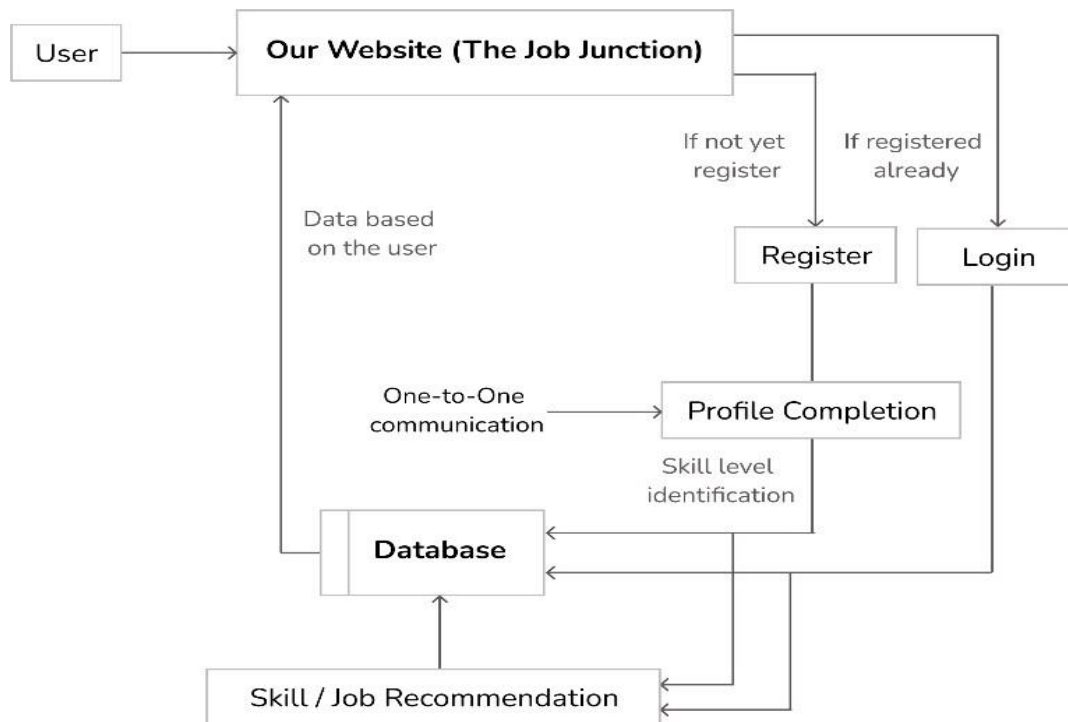
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration via Form on our website.
FR-2	Profile Completion	User will create their profile with their basic personal details, we will call them in person to know their educational background, skill levels, and their goals to provide them with better service. The gathered details will be stored and displayed in their profile section.
FR-3	User Login	Users can log in with either their Email or Phone number.
FR-4	Job / Skill Recommendation	Based on their profile, we will recommend them appropriate jobs with courses to learn if needed. Will be notified through Email and Our Website.

4.2 Non-Functional Requirement

FR No.	Non-Functional Requirement	Description
NFR-1	Usability	Right now, we are focusing on the target users (fresher and college students) without any special considerations. We will definitely make it usable for all kinds of people in the future.
NFR-2	Security	Basic form validations will be performed to avoid harmful injections and also to get meaningful information.
NFR-3	Reliability	We are planning to serve 1000 users at a time in the initial stages.
NFR-4	Performance	Mobile responsive website.
NFR-5	Availability	Planning to provide customer support during office hours 9-5.
NFR-6	Scalability	In addition to this, we can provide career guidance programs, job fairs, native language support, and premium services like personalized mentoring, coaching, and so on in the future.

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technology Architecture

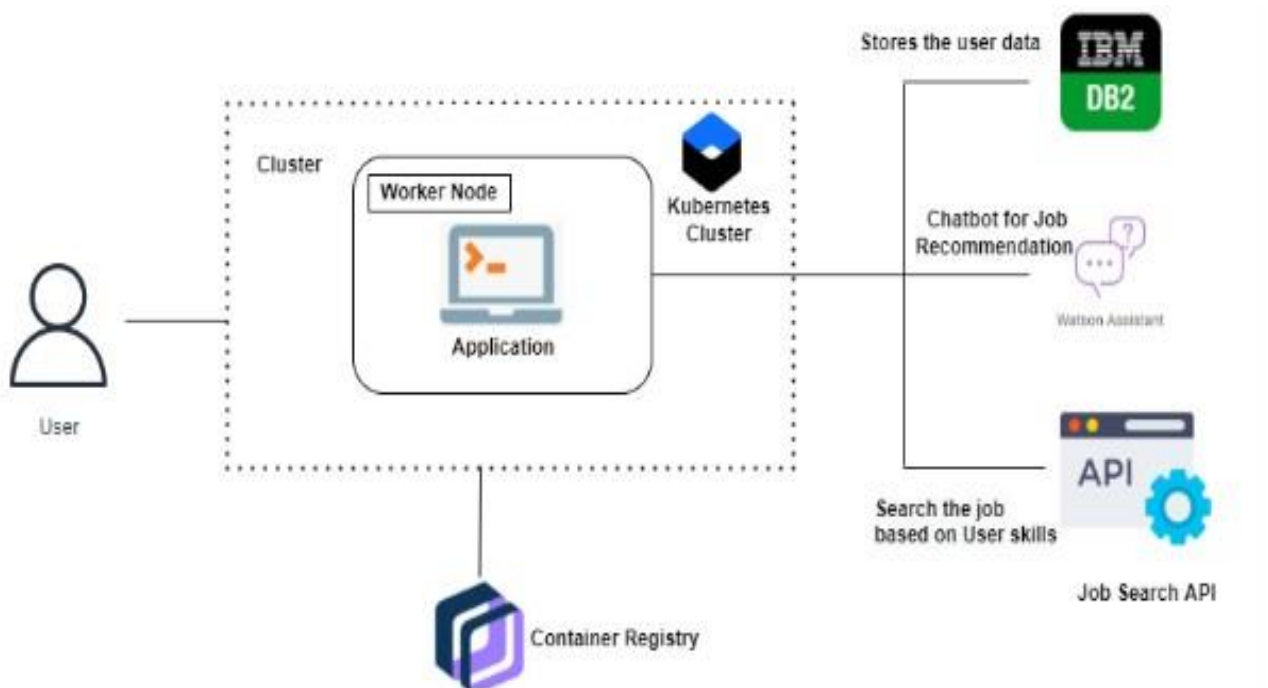


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Front-end of our platform was Developed using these technologies.	HTML, CSS (flexbox) and JavaScript
2.	Application Logic-1	Displaying the web pages Based on the criteria.	Python
3.	User Onboarding	Welcoming users inside the website using IBM Watson Assistant.	IBM Watson Assistant
4.	Database	Queries for fetching the required details.	MySQL.
5.	Cloud Database	Database Service on Cloud	IBM DB2.
6.	File Storage	File storage requirements	IBM Block Storage.
7.	Infrastructure (Server / Cloud)	Application Deployment.	Kubernetes.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Website developed using These open-source frameworks.	Flask and Bootstrap
2.	Security Implementations	To ensure the valid details are getting from the user.	Input validations and encryption of user's data.
3.	Scalable Architecture	Can be easily scalable, as we are dealing with IBM technologies for implementing our solutions.	IBM DB2, IBM Watson Assistant, and Kubernetes (combined architectures).
4.	Availability	Justify the availability of applications (e.g. use of load balancers, distributed servers, etc.)	None
5.	Performance	By writing optimized code and different techniques, we are trying to provide better and faster performance.	None

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user & Web user)	Registration	USN-1	As a user, I can register for the application by entering my name, email, phone number, and password.	I can access to complete my profile.	High	Sprint-1
		USN-2	As a user, I can register through <u>Chatbot</u> .	I can access to complete my profile.	Low	Sprint-4
	Login	USN-1	As a user, I can log into the application by entering email & password.	I can access to see job openings for my skill levels.	High	Sprint-1
	Notification	USN-1	As a registered user, I can get notification for jobs / courses recommendations based on my skills through calls and whatsapp.	I can apply for the jobs or learn the recommended courses.	High	Sprint-2
		USN-2	As a registered user, I can get notification for jobs / courses recommendation through Email.	I can apply for the jobs or learn the recommended courses.	Medium	Sprint-3
	Recommendation	USN-1	As a user, I can view job recommendations or courses to improve myself in the website.	I can apply for jobs or learn skills based on the recommendation.	High	Sprint-1
Customer Care Executive	Contact registered users	CCESN-1	As a customer care executive, I can talk to the registered users to know their skill levels and their interest.	I can complete their profile for job / skill recommendations.	High	Sprint-1
Administrator	Monitoring the user on boarding	ASN-1	As an administrator, I monitor whether a user complete his profile once he registered.	I can make sure each and every user has a complete profile for further processes.	High	Sprint-1

6.PROJECT PLANNING

6.1 Sprint Planning & Estimation

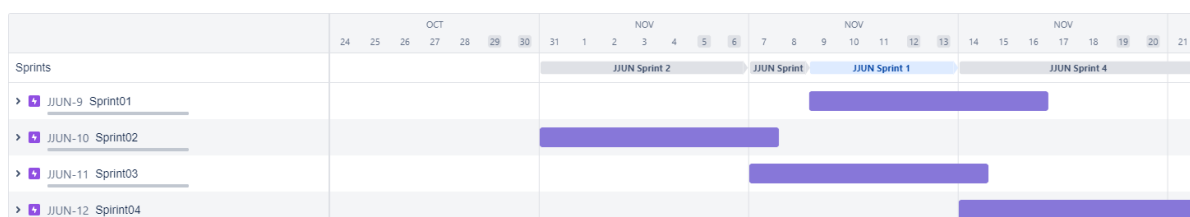
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sakthivel Sathish
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	Sanjay kumar Swathi
Sprint-2		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sakthivel Swathi
Sprint-3		USN-4	As a user, I can register for the application through Gmail	I can receive confirmation email & click confirm	Medium	Sakthivel Swathi
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email & password	I can access my account / dashboard	High	Sakthivel
Sprint-2	Dashboard	USN-6	Create a model set that contains those models, then assign it to a role.	Assign that group to the appropriate roles on the Roles page	High	Swathi Sathish
Sprint-4	Identity-Aware	USN-7	Open, public access, User-authenticated access, Employee-restricted access.	Company public website. App running on the company intranet. App with access to customer private information.	High	Sakthivel Swathi
Sprint-1	Communication	USN-8	A customer care executive is a professional responsible for communicating the how's and why's regarding service expectations within a company.	For how to tackle customer queries.	Medium	Sathish Sanjay kumar
Sprint-3	Device management	USN-9	You can Delete/Disable/Enable devices in Azure Active Directory but you cannot Add/Remove Users in the directory.	Ease of use.	Medium	Sakthivel Swathi

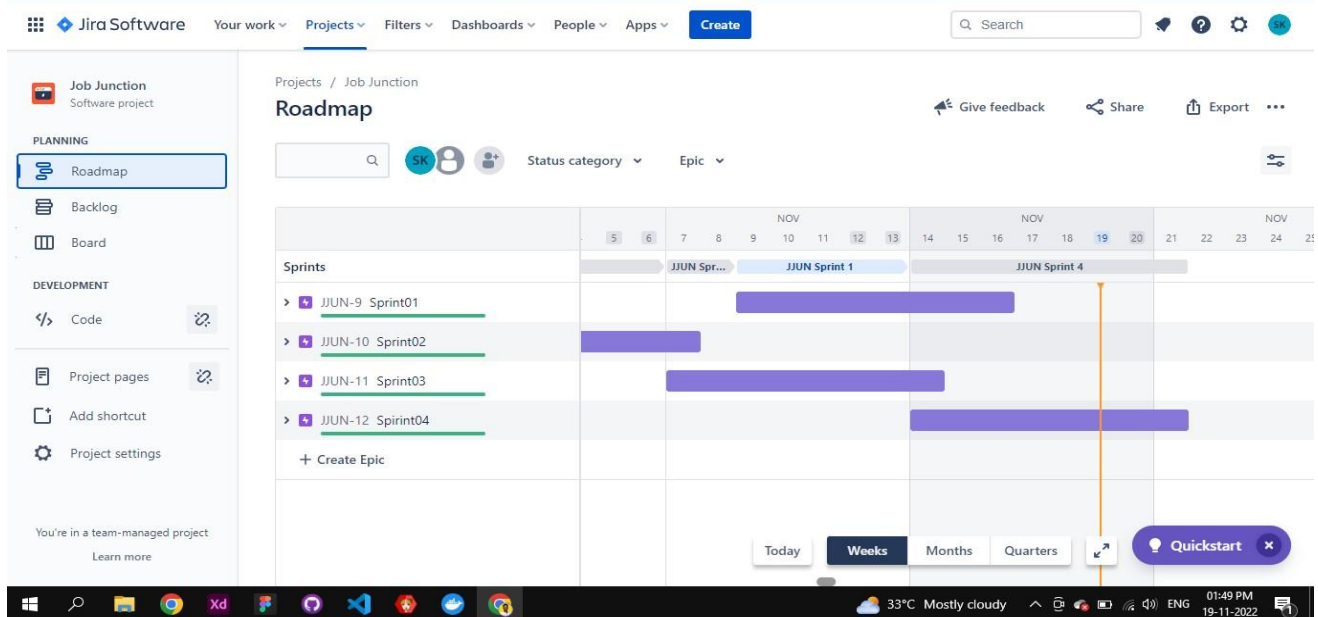
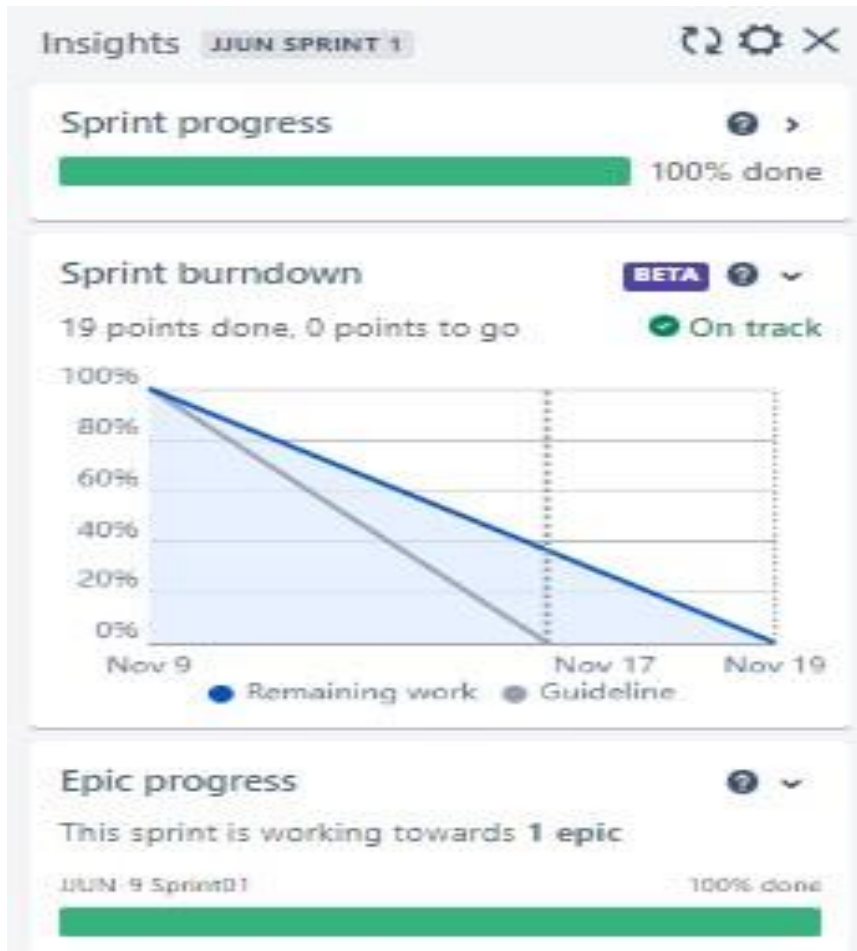
6.2 Sprint Delivery Schedule

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	19	6 Days	24 Oct 2022	29 Oct 2022	19	16 Nov 2022
Sprint-2	01	6 Days	31 Oct 2022	05 Nov 2022	01	17 Nov 2022
Sprint-3	01	6 Days	07 Nov 2022	12 Nov 2022	01	18 Nov 2022
Sprint-4	01	6 Days	14 Nov 2022	19 Nov 2022	01	19 Nov 2022

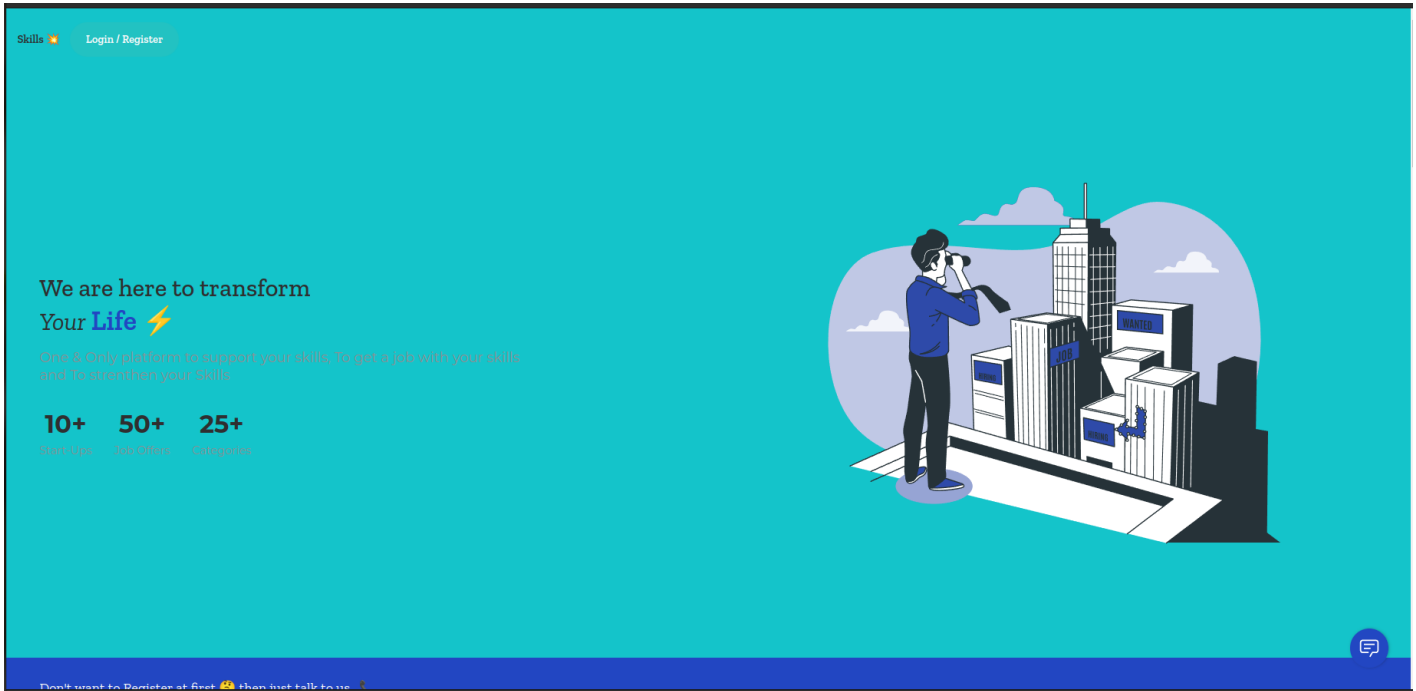
Reports from JIRA





7.CODING & SOLUTIONING

7.1 Chatbot Integration Code

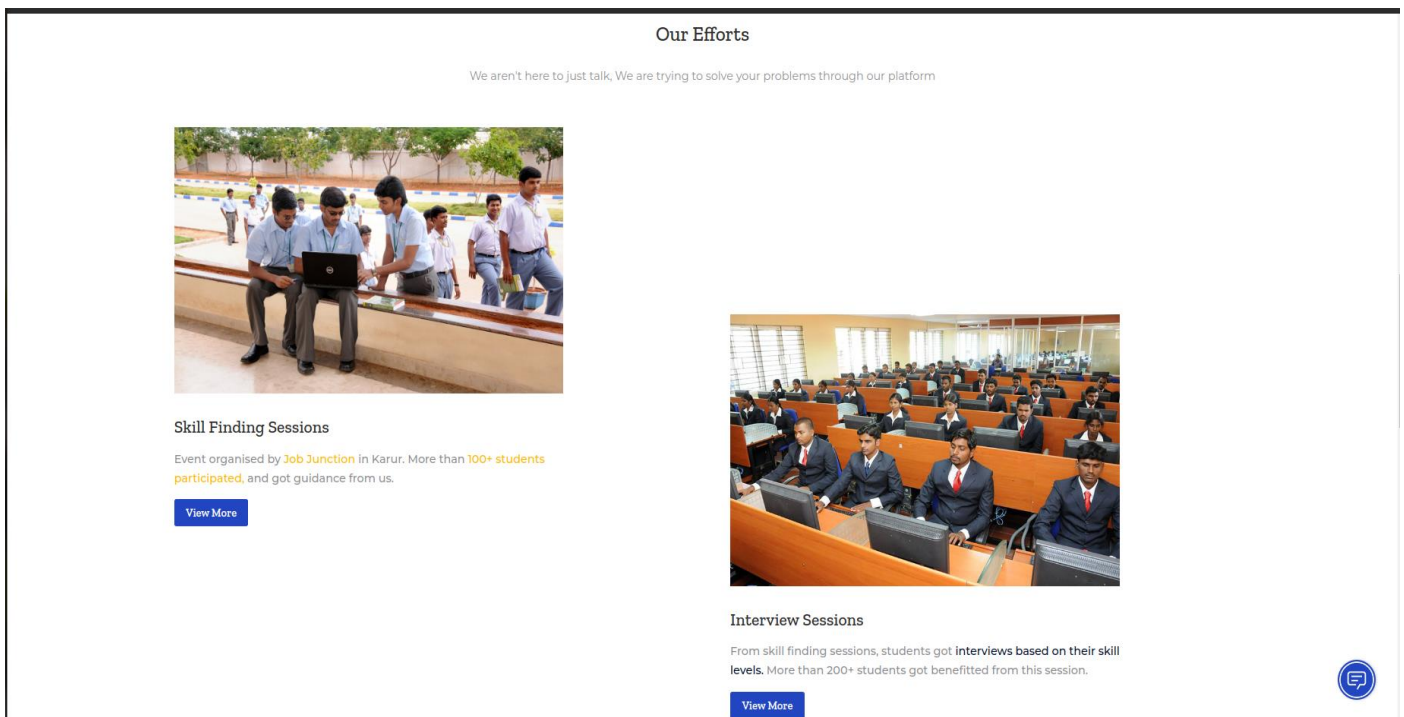
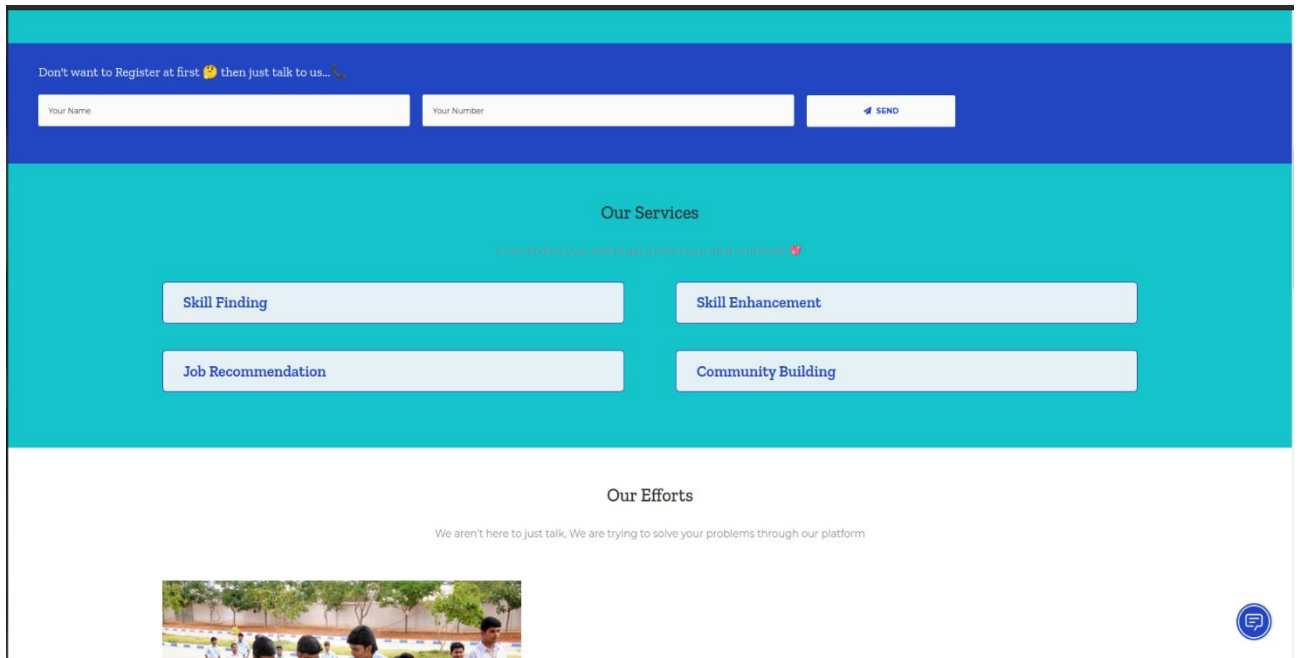


```
<script>
window.watsonAssistantChatOptions = {
  integrationID: "c9539449-0fd4-4aca-8b8b-3a6c109a71fa", // The ID of this integration.
  region: "jp-tok", // The region your integration is hosted in.
  serviceInstanceID: "39284af4-f9b4-4159-be01-8ca534e21855", // The ID of your service
instance.
  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 FLASH FRAMEWORK


Responsive Website

Desktop View



Dynamic Application

Registration Form

Skills  [Login / Register](#)

Register to move ahead in your life

Fill your basic information. We will contact you further guidance till job recommendation based on your skill

Your Name

Your Email

Your Mobile Number

Your Password


Your Password

REGISTER

Already have an account ?

[Let's Login](#)

Login After Registration

Skills  [Login / Register](#)

Login to Pursuing your goals

You are at right time. Start learn and apply for jobs

Your Email

Your Password

LOGIN

Don't have an account? Register

Profile After Login

Skills 🌟

Your Profile 🧑

Welcome Back!

SAKTHIVEL

shanmugamsakthivel442@gmail.com

987547344

[Explore Jobs 📄](#)

LOGOUT

Jobs After Login

Skills 🌟

Your Profile 🧑

Best Guidance For You 📁

Hand picked recommendations for you based on your skillset

One Step ahead 🚀

We recommend you the skill enhancement guidance, once we know about your skills.
Don't hesitate or fear to call us.

Call Us 📞

Do you want Skilled Employees. Fill this form. Our team will get back you 📞

Your Name

Your Company Name

Employee Role

Your Email

Your Number

SEND

7.3 Database Schema

IBM Db2 on Cloud

Load DataLoad HistoryTablesViewsIndexesAliasesMQTsSequencesApplication objects

Find schemas or tables

Refresh

Schemas

<input checked="" type="checkbox"/>	Name	Type	Tables
<input checked="" type="checkbox"/>	ZTC89378	User	6

Total: 1, selected: 1

Tables

New table

☐

Name

Schema

Properties

☐

CONNECTWITHUS

ZTC89378

...

☐

CONTACT

ZTC89378

...

☐

HIRERFORM

ZTC89378

...

☐

REGISTER

ZTC89378

...

☐

SIGNUP

ZTC89378

...

☐

STUDENTS

ZTC89378

...

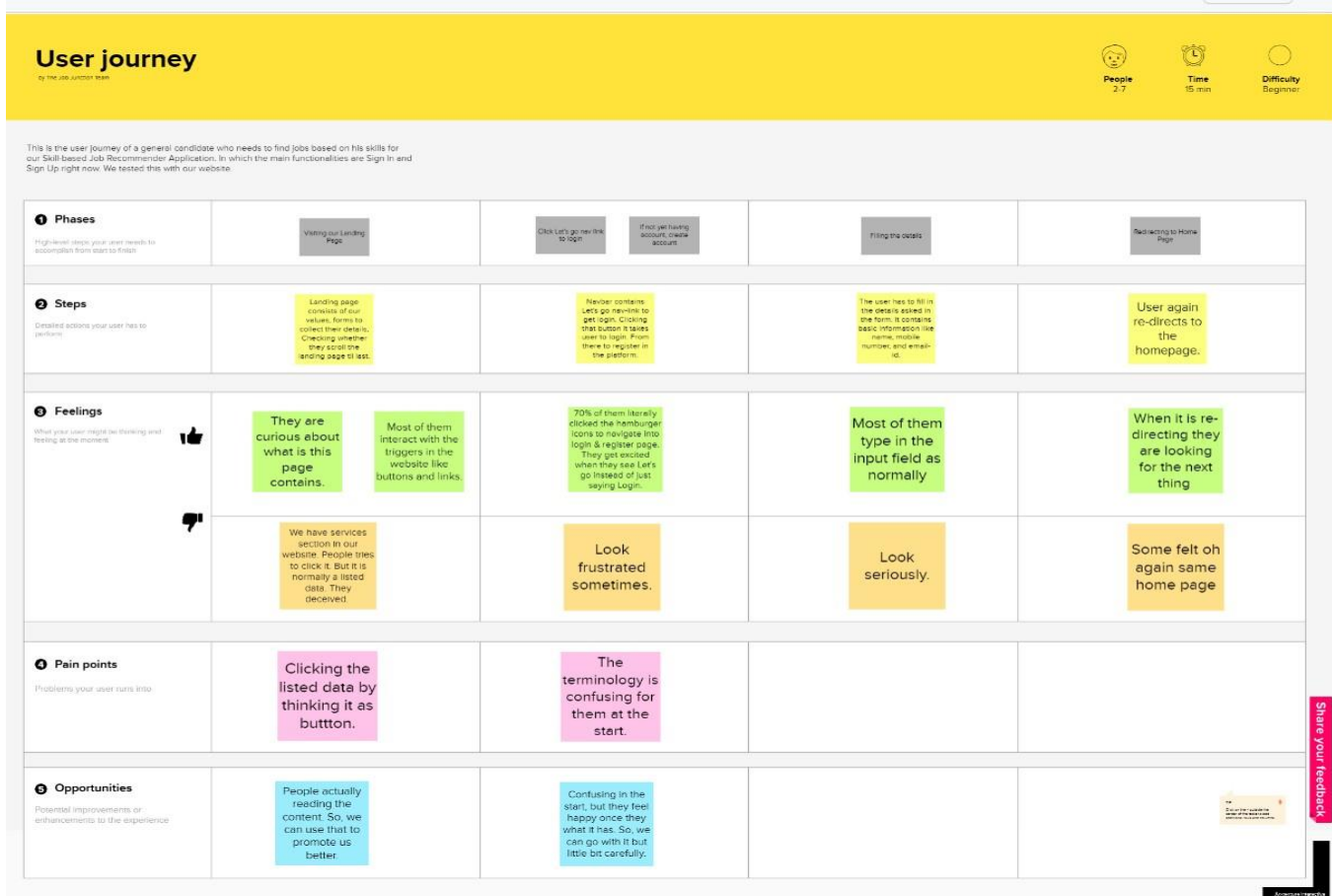
Total: 6, selected: 0

8.TESTING

8.1 TEST CASES

- 1. Website functionality testing :** It includes Outbound links, Internal links correctness, There are no links leading to the same page, There are no broken links.
- 2.Usability testing :** It includes Navigation testing and Content testing.
- 3.UI (User Interface) testing :** It includes Testing with different screen resolutions and Design elements evaluation: layout, colors, fonts, font sizes, labels, text boxes, text formatting, captions, buttons, lists, icons, and links.
- 4.Performance testing :** This is carried out by Lighthouse in the browser.

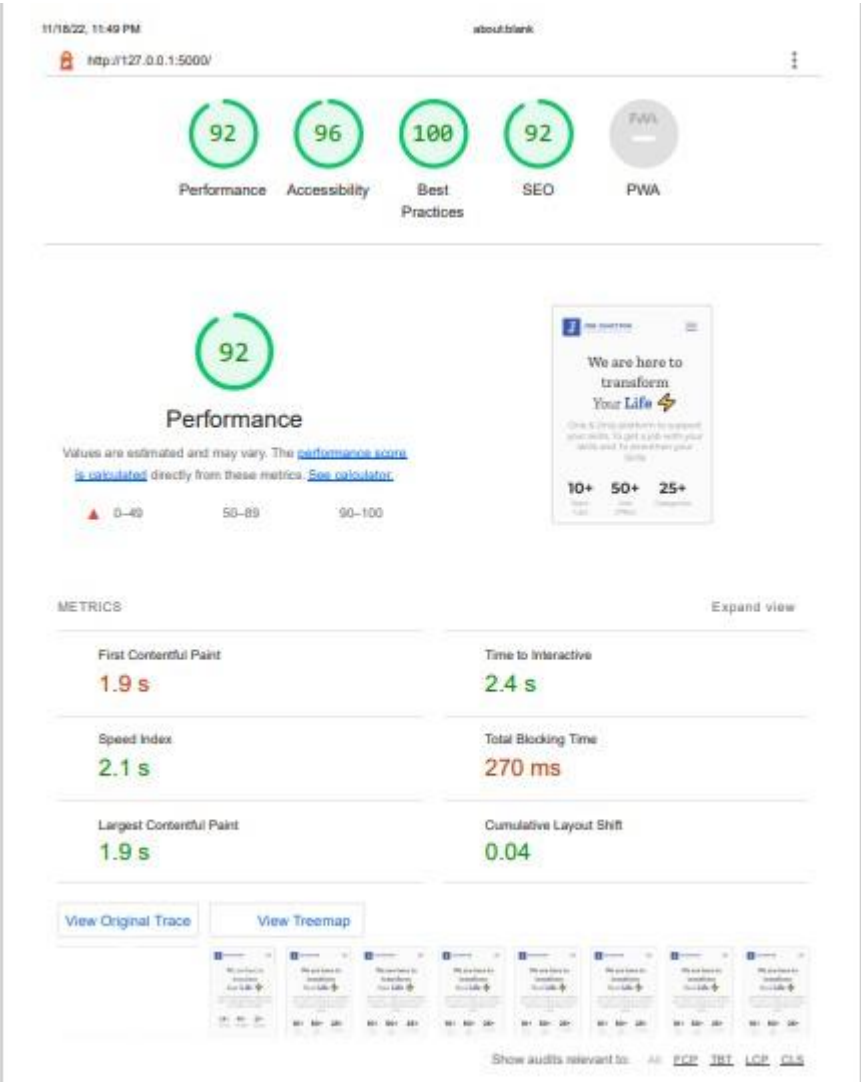
8.2 User Acceptance Testing



We performed this User journey mapping before starting development of our project. We did this with our end users (students of our college). Based on the observations, we did this application.

9. RESULTS

9.1 Performance Metrics:



10. ADVANTAGES & DISADVANTAGES

10.1 Advantages:

1. Right job/skill recommendations based on their interest, skill level, and expertise in a particular domain.
2. Achieving a Win-Win policy for both job seekers and providers.

10.2 Disadvantages:

1. Time consuming process.
2. Tedious process.

11.CONCLUSION

We did our best to our capacity. We did each and every process and task after learning. We actually started it from scratch and come to this level. We learned Project planning and execution along with the development languages like HTML, CSS, Bootstrap, Flask, IBM cloud DB2, Docker, and so on. We are actually planning to take this project idea for our start-up. The above-produced documents are all done by us. Yet to improvise and complete a lot with this project. We strongly believe this idea will great impact on people's life.

12.FUTURE SCOPE

We found the core problem among job seekers which we mentioned in the above existing problem section. Now, we are trying to solve it by simply finding users' skill levels in a more human-centered way and recommending jobs based on their category at all levels. We believed that this idea can be scalable. Because once we achieve/find an implementation approach to our above-mentioned primary task then,

1. It can be expandable by providing offline support along with online.
2. End-to-end support for skills identification, enhancement, job support, and more.

13.APPENDIX

Source Code

----TEMPLATES----

#Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Job Junction</title>
  <link rel="icon" type="image/gif/png" href="../../static/images/job_junction-
logo-head.png">
  <script src="https://kit.fontawesome.com/7bb9495e7b.js"
crossorigin="anonymous"></script>
  <link rel="stylesheet" href="{{ url_for('static',
filename='css/indexstyles.css') }}">
  <script src="{{ url_for('static', filename='index.js') }}" defer></script>

  <style>

    /*#body{
      background-color: #2246c247;
    }*/
```

```

.modal{
    text-align: center;
}

.modal_bottom{
    background-color: #2246C2;
    padding:24px;
    display: inline-block;
    border-radius: 5px;
}

.modal_bottom p{
    font-family: 'Zilla Slab';
    font-style: normal;
    font-weight: 400;
    font-size: 20px;
    line-height: 38px;
    display: flex;
    align-items: center;
    color: #F8F8F8;
}

.modal_send{
    background: #FBFBFB;
    border-radius: 3px;
    margin-top: 12px;
    border: none;
    padding:16px 8px;
    outline:none;
    margin-right: 16px;
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 700;
    font-size: 12px;
    line-height: 128.69%;
    text-align: center;
    color: #2246C2;
}
</style>

<script>
    window.watsonAssistantChatOptions = {
        integrationID: "c9539449-0fd4-4aca-8b8b-3a6c109a71fa", // The ID of this
integration.
        region: "jp-tok", // The region your integration is hosted in.
        serviceInstanceID: "39284af4-f9b4-4159-be01-8ca534e21855", // 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>

</head>
<body id="body">

    <div class="menu_icon" id="menu">
        <div class="first_div"></div>
        <div class="second_div"></div>
        <div class="third_div"></div>
    </div>

    <header>
        <div class="brand_box">
            
        </div>
        <nav>
            <ul class="nav_links" id="nav_links">
                <li class="nav_link"><a href="/jobs">Skills *</a></li>
                {% if session['email'] %}
                <li class="nav_link special_link"
id="special_link"><a href="/profile" id="contact_link">Your Profile
👤</a></li>
                {% else %}
                <li class="nav_link special_link"
id="special_link"><a href="/register" id="contact_link">Login /
Register</a></li>
                {% endif %}
            </ul>
        </nav>
    </header>

    <div class="hero">
        <div class="_hero hero_left">
            <div>

```

----STATIC FILES----

----CSS FILES----

#growthzonestyles.css

```
@import
url('https://fonts.googleapis.com/css2?family=Zilla+Slab:ital,wght@0,300;0,400
;0,500;0,600;0,700;1,300;1,400;1,500;1,600;1,700&display=swap');
@import
url('https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,400;0,500
;0,600;0,700;0,800;0,900;1,400;1,500;1,600;1,700;1,800;1,900&display=swap');

*{
  margin:0;
  padding:0;
  box-sizing:border-box;
  list-style-type: none;
}

/*body*/
body{
  background: #FBFBFB;
  font-family: 'Zilla Slab', serif;
  font-family: 'Montserrat', sans-serif;
}

/*Top - Header*/
header{
  width:100%;
  padding:20px 16px;
  display: flex;
  justify-content:space-between;
  align-items: center;
}

/*HeaderContainer*/
.brand_box{
```

```

        display: flex;
        font-size: 20px;
        width:30%;
    }

    /*Header-brandbox-Logo*/
    .brand_box img{
        width:55%;
    }

    /*header-navbar-container*/
    .nav_links{
        display: flex;
        align-items: center;
    }

    /*header-navbar-link-anchor*/
    .nav_links .nav_link a{
        margin-right: 15px;
        text-decoration: none;
        font-family: 'Zilla Slab';
        font-style: normal;
        font-weight: 600;
        font-size: 16px;
        line-height: 34px;
        color: #2F2F30;
    }

    /*header-navbar-link-container*/
    .nav_links .special_link{
        background: #2246C2;
        border-radius: 32px;
        padding:8px;
        text-align: center;
    }

    /*header-navbar-link-anchor-SPECIAL*/
    .nav_links .special_link a{
        color:#FBFBFB;
        margin-left:15px;
    }

    /*hamburger icon container*/
    .menu_icon{
        display: none;
        position: absolute;
        cursor:pointer;
        top: 5%;
    }

```



```

    right: 26px;
    z-index:20;
}

/*hamburger-container-divs*/
.menu_icon div{
    display:block;
    width:26px;
    height:4px;
    margin-bottom: 4px;
    background-color: #9d9bbe;
    border-radius: 10px;
}

/*topsection container*/
.hero{
    display: flex;
    flex-direction: row;
    align-items:center;
    padding:32px;
    position: relative;
}

/*input tag*/
input{
    margin-top: 16px;
    width:30%;
    background: #FBFBFB;
    border: none;
    border-radius: 3px;
    padding:16px;
    outline:none;
    margin-right: 16px;
}

/*target button*/
.button_send{
    width:12%;
    margin-top: 16px;
    background: #FBFBFB;
    border-radius: 3px;
    border: none;
    padding:16px 8px;
    outline:none;
    margin-right: 16px;
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 700;

```

```

    font-size: 12px;
    line-height: 128.69%;
    text-align: center;
    color: #2246C2;
}

/*input placeholder*/
input::placeholder{
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 400;
    font-size: 12px;
    text-align: left;
    color: #2F2F30;
}

/*target button icon*/
.icon{
    margin-right: 8px;
}

/*section containers - head*/
.ourgrowths .header .head{
    font-family: 'Zilla Slab';
    font-style: normal;
    font-weight: 500;
    font-size: 28px;
    line-height: 58px;
    color: #2F2F30;
    text-align: center;
}

/*section containers - description*/
.ourgrowths .header .description{
    margin: 10px 16%;
    text-align: center;
    font-family: 'Montserrat';
    font-style: normal;
    font-weight: 400;
    font-size: 16px;
    line-height: 128.69%;
    text-align: center;
    color: #8F8F92;
}

/*ourgrowths - body container*/
.ourgrowths .body{
    margin-top: 12px;

```

---JAVASCRIPT FILES----

#index.js

```
let menu = document.getElementById("menu");
let links = document.getElementById("nav_links");
let body = document.getElementById("body");

menu.addEventListener("click", ()=>{
    links.classList.toggle("nav_responsive");
    links.classList.toggle("navbac");
    body.classList.toggle("bacdrop");
});
```

----PYTHON FILES---

#app.py

```
from turtle import st
from flask import Flask, render_template, request, redirect, url_for, session
from markupsafe import escape
import requests
import json

import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=55fbc997-9266-4331-afd3-888b05e734c0.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31929;SECURITY=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=ztc89378;PWD=HSYKGIqGBcRZVRHK",'','')
print("Connected Successfully !")

app = Flask(__name__)

app.secret_key = 'Done ehhe'

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

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

@app.route('/login')
def login():
    return render_template('login.html')
```

```

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

@app.route("/registerrec", methods = ['POST', 'GET'])
def registerrec():
    msg=''
    if request.method == 'POST':

        name = request.form['name']
        number = request.form['number']
        email = request.form['email']
        password = request.form['password']
        cpassword = request.form['cpassword']

        sql = "SELECT * FROM register WHERE email =?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        ibm_db.execute(stmt)
        account = ibm_db.fetch_assoc(stmt)
        print(account)

        if account:
            return render_template('login.html', msg="You are already a
member, please login using your details")
        else:
            insert_sql = "INSERT INTO register VALUES (?, ?, ?, ?, ?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prepare_stmt, 1, name)
            ibm_db.bind_param(prepare_stmt, 2, number)
            ibm_db.bind_param(prepare_stmt, 3, email)
            ibm_db.bind_param(prepare_stmt, 4, password)
            ibm_db.bind_param(prepare_stmt, 5, cpassword)
            ibm_db.execute(prepare_stmt)

            return render_template('login.html', msg="Registered
successfully..login to continue")
#Login

@app.route("/loginrec", methods =['POST','GET'])
def loginrec():
    smsg = ''
    if request.method == 'POST' and 'email' in request.form and 'password' in
request.form:
        email = request.form['email']
        password = request.form['password']
        if((email and password) is not None ):

```

```

        sql = "SELECT * FROM register WHERE email = ?"
        stmt = ibm_db.prepare(conn, sql)
        ibm_db.bind_param(stmt,1,email)
        prep_stmt = ibm_db.execute(stmt)
        dicto = ibm_db.fetch_assoc(stmt)
        while(dicto != False):
            res1 = dicto["EMAIL"]
            res2 = dicto['PASSWORD']
            res3 = dicto['NAME']
            res4 = dicto['NUMBER']

            if(res1 == email and res2 == password):
                session['email'] = res1
                session['pass'] = res2
                session['name'] = res3
                session['number'] = res4
                return render_template('profile.html')
            else:
                return render_template('login.html', smsg = 'Incorrect
username / password !')
            else:
                return render_template('register.html', smsg = 'Not yet
registered')
            else:
                return render_template('login.html', smsg = 'Fill all the
details')
            else:
                return render_template('login.html')

#contact

@app.route("/contactrec",methods = ['POST', 'GET'])
def contactrec():
    cmsg=''
    if request.method == 'POST':

        Name = request.form['name']
        MobileNumber = request.form['number']

        insert_sql = "INSERT INTO contact VALUES (?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prepare_stmt, 1, Name)
        ibm_db.bind_param(prepare_stmt, 2, MobileNumber)
        ibm_db.execute(prepare_stmt)

        return render_template('index.html', cmsg="We will contact you soon"

```

#sendmail.py

```
import configparser
import os
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
from sendmail import SendGridAPIClient
from sendgrid.helpers.mail import Mail

config = configparser.ConfigParser()
config.read("config.ini")
```

```

def SendmailUsingSendGrid(API,from_email,to_email,subject,html_content):
    if API != None and from_email != None and len(to_email)>0 :
        message = Mail(from_email,to_email,subject,html_content)
        try:
            sg = SendGridAPIClient(API)
            response = sg.send(message)
            print(response.status_code)
            print(response.body)
            print(response.headers)
        except Exception as e:
            print(e.message)

try:
    settings = config["SETTINGS"]
except:
    settings = {}

API = settings.get("APIKEY",None)
from_email = settings.get("FROM",None)
to_email = settings.get("TO","")

print(API)

subject = "Sample test message"
html_content = "Message Successfully sent through Python Sendgrid"

SendmailUsingSendGrid(API,from_email,to_email,subject,html_content)

```

-----TEXT FILE----

#requirements.txt

```

flask
ibm_db

```

----DOCKER FILE----

```

FROM python:3.10.6
WORKDIR /app
COPY requirements.txt ./
RUN pip install -r requirements.txt
COPY . .
EXPOSE 5000
CMD ["python", "./app.py"]

```

----INI FILE----

#config.ini

```
["SETTINGS"]
APIKEY = SG.iYENC4V3Rg-EmgxAgr8vGw.QPtuntiaOHhcYI85o-kS3suDoR2muCFwHGZzoCNV1Q
FROM = govaaece@gmail.com
TO = session['email']
```

GitHub

GitHub Link 1 : <https://github.com/IBM-EPBL/IBM-Project-129-1665375319>

GitHub Link 2 : <https://github.com/IBM-EPBL/IBM-Project-129-1658212282>

Project Demo Link

Googledrive link :

https://drive.google.com/file/d/1jR1a3uQ1gkHcNvPJFRSA8B7_Wlj3LgbN/view?usp=share_link