# 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 | SIGNATURE | SIGNATURE |
| Dr .N.S.NITHYA | Dr. V. BABY ANITHA | Dr. P.SENTHILKUMAR |
| FACULTY MENTOR | FACULTY EVALUATOR | PRINCIPAL AND HEAD |
| Associative Professor | Associative Professor | Professor |
| Department of Computer Science and Engineering | Department of Computer Science and Engineering | Department of Computer Science and Engineering |

# 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 withProject 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 theuser Pains & Gains |
| | | 4 | 12 - 17 Sept 2022 | Listing of the ideas using brainstorming session |
| 3 | Project Design Phase -I (ProposedSolution, 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, CustomerJourney, 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 TechnologyArchitecture |
| 5 | Project Planning Phase (Milestones & Tasks, Sprint Schedules ) | 9 | 17 - 22 Oct 2022 | Preparing Milestone & Activity List, SprintDelivery 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<br>1.1 PROJECT OVERVIEW<br>1.2 PURPOSE | 5 |
| 2 | LITERATURE SURVEY<br>2.1 EXISTING PROBLEM<br>2.2 REFERENCE<br>2.3 PROBLEM STATEMENT | 5 |
| 3 | IDEATION & PROPOSED SOLUTION<br>3.1 EMPATHY MAP CANVAS<br>3.2 IDEATION AND BRAINSTORMING<br>3.3 PROPOSED SOLUTION<br>3.4 PROBLEM SOLUTION FIT<br>3.5 SOLUTION ARCHITECTURE | 8 |
| 4 | REQUIREMENT ANALYSIS<br>4.1 FUNCTIONAL REQUIREMENTS<br>4.2 NON FUCTIONAL REQUIREMENTS | 13 |
| 5 | PROJECT DESIGN<br>5.1 DATA FLOW DIAGRAMS<br>5.2 SOLUTION AND TECHNICAL ARCHITECTURE<br>5.3 USER STORY | 14 |
| 6 | PROJECT PLANNING & SCHEDULING<br>6.1 SPRINT PLANNING AND ESTIMATION<br>6.2 SPRINT DELIVERY SCHEDULE | 17 |
| 7 | CODING & SOLUTIONING<br>7.1 CHATBOT INTEGRATION CODE<br>7.2 FLASK FRAMEWORK<br>7.3 DATA BASE SCHENMA | 19 |
| 8 | TESTING<br>8.1 TEST CASES<br>8.2 USER ACCAPTANCE TESTING | 25 |
| 9 | RESULTS<br>9.1 PERFORMANCE METRICES | 27 |
| 10 | ADVANTAGES & DISADVANTAGES<br>10.1 ADVANTAGES<br>10.2 DISADVANTAGES | 28 |
| 11 | CONCLUSION | 29 |
| 12 | FUTURE SCOPE | 29 |
| 13 | APPENDIX | 29 |

# 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 orthe 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 peoples 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.

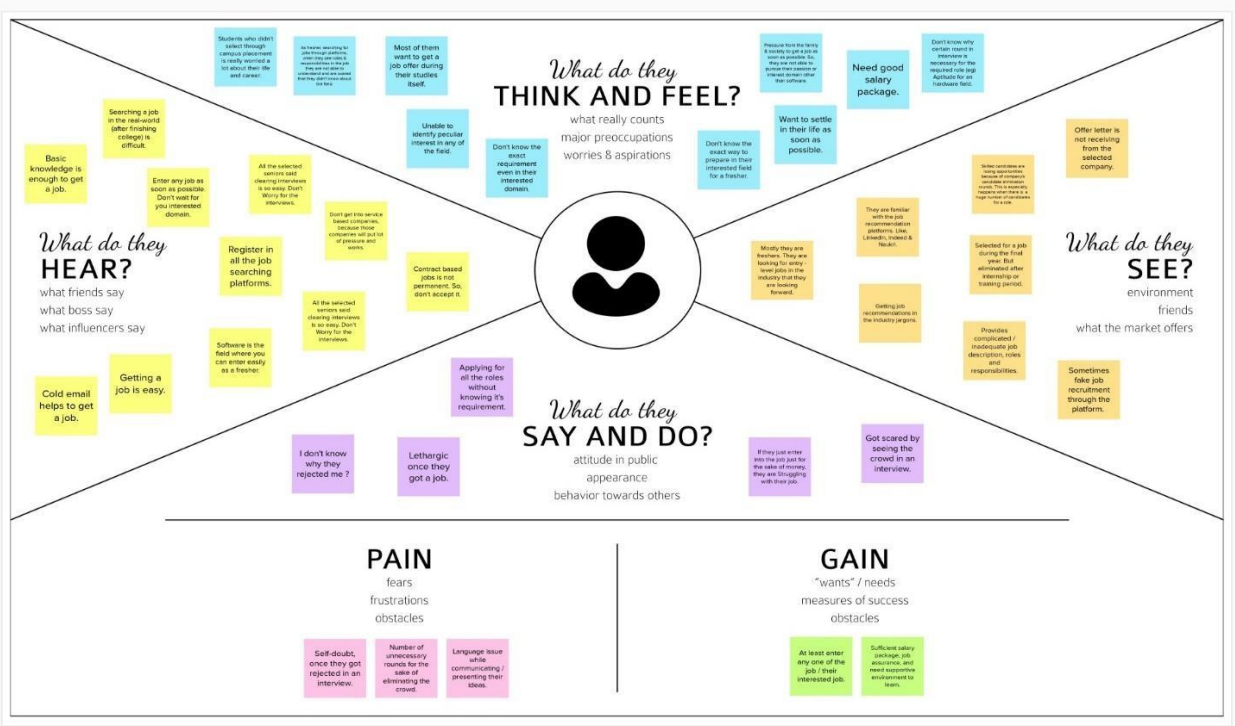
**2.2 References:**

| 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 sendingnotification to the students.Thus proper job recommendations are provided to the students. | 1. They use the more logical wayapproach.<br>2. Getting completely detailed infromation -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 similarapproach that wehave now for recommendation, but ended up withthe 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<br><br>March 2018. | In this paper, proposed a framework for job recommendation task. This framework facilitates the understanding of job recommendation processas well as it allows theuse of a variety of textprocessing 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-Otaibi1*, Mourad Ykhlef2<br><br>Aug 2012 | This survey shows that several approaches for job recommendationhave 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 realizedlot to learn.<br>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:

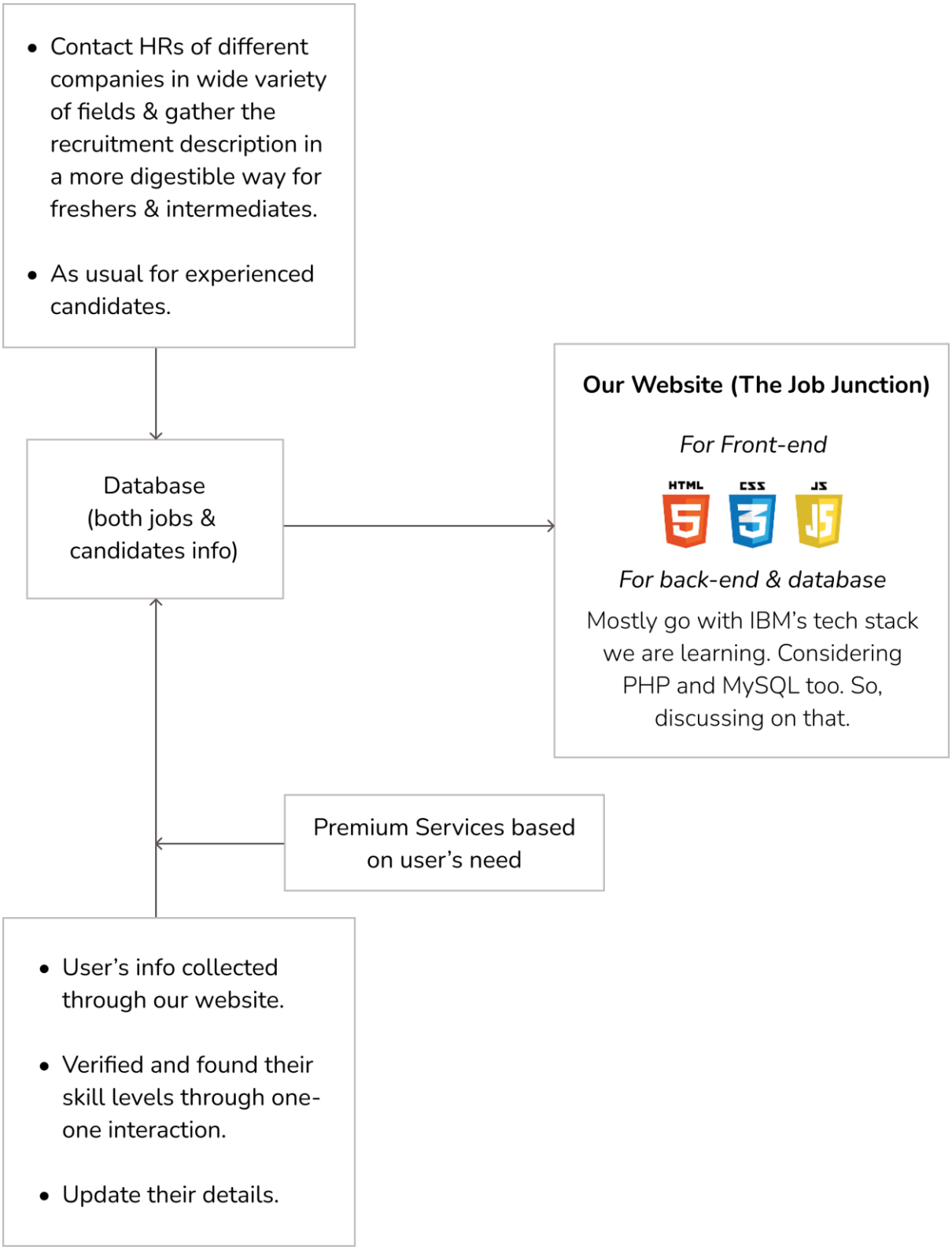
### 3.3 Proposed Solution:

| 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 | 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.** Bythis, 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 findthe right skill level for the people. In addition, we can **get more insights** about them. This creates a more positive impact and customer<br>satisfaction in using our service. |
| 5. | Business Model (Revenue Model) | 1. **Advertisement** commissions by allowing skill development companies and job-providing agencies to promote their business to reach their potential users.<br>2. **Premium services** like mentor support, exclusive courses, etc… |
| 6. | Scalability of the Solution | 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,<br>**1.** It can be expandable by **providing offline support along with online.**<br>2. **End-to-end support** for both skills identification, enhancement, job support, and more. |

## 3.4 Problem Solution Fit:

### 3.5 Solution Architecture:



## 4. REQUIREMENT ANALYSIS

### 4.1 Functional Requirement

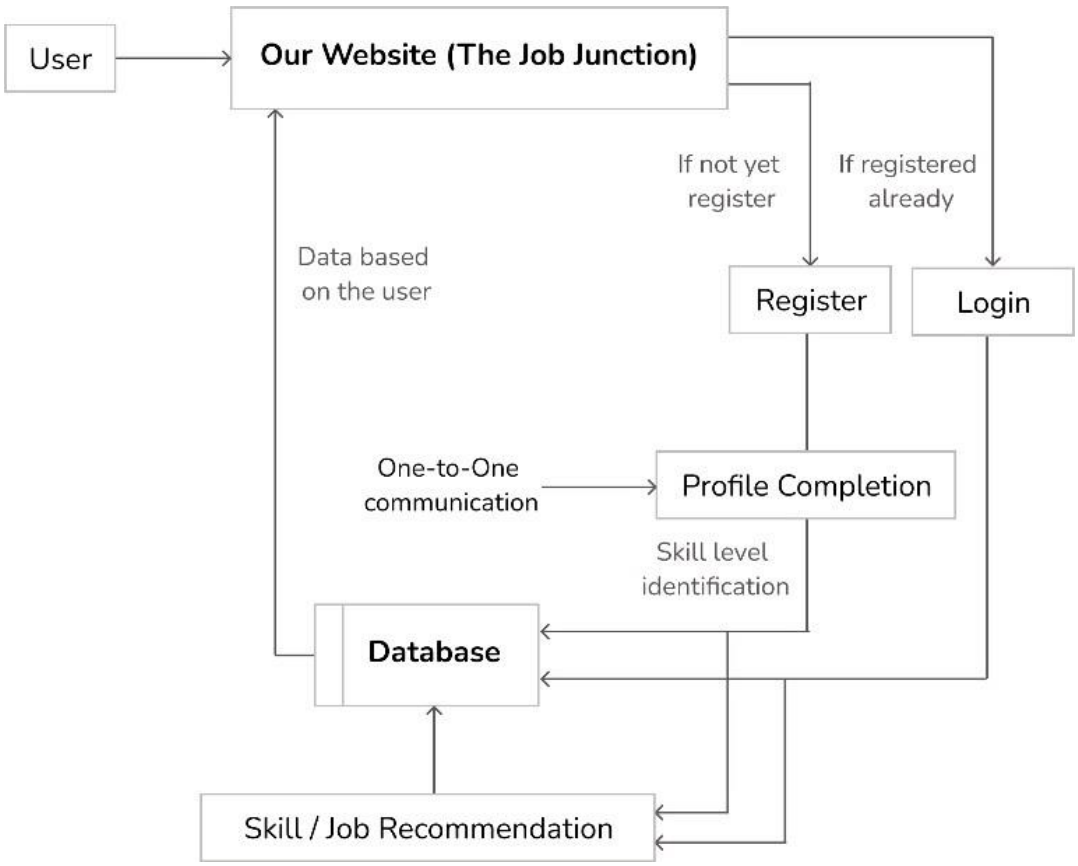
| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration via Form on our website. |
| FR-2 | Profile Completion | User will create their profile with their basic personal details, we will call them in person to know their educational background, skill levels, andtheir 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

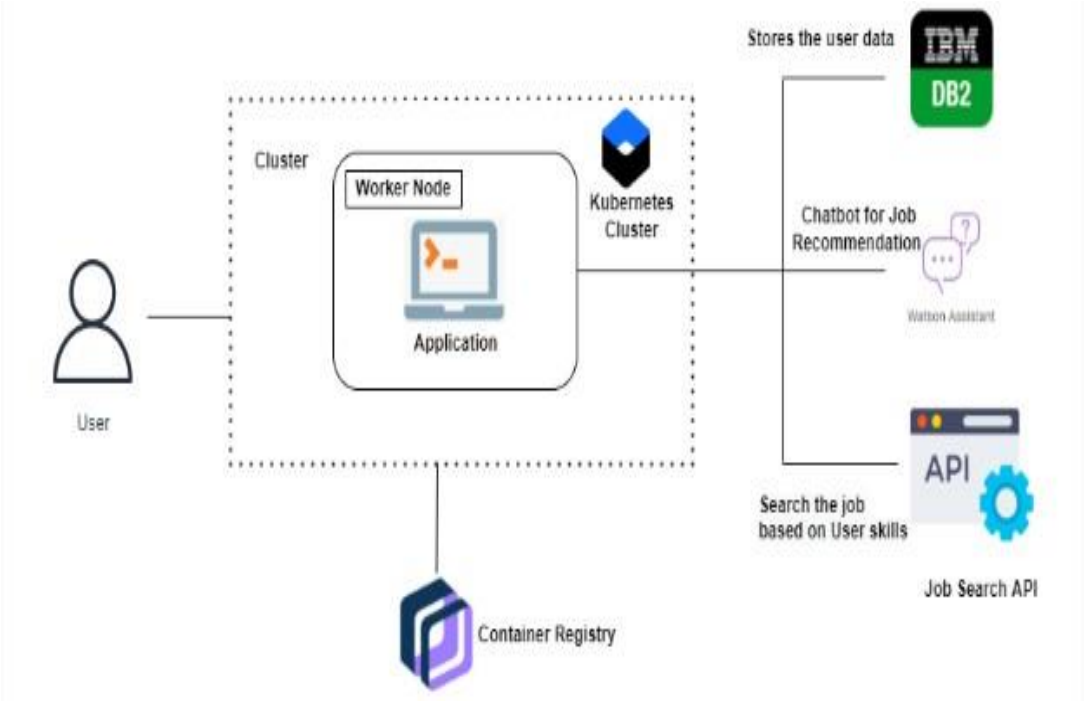


**Table-1 : Components & Technologies:**

| 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 Chatbot. | 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

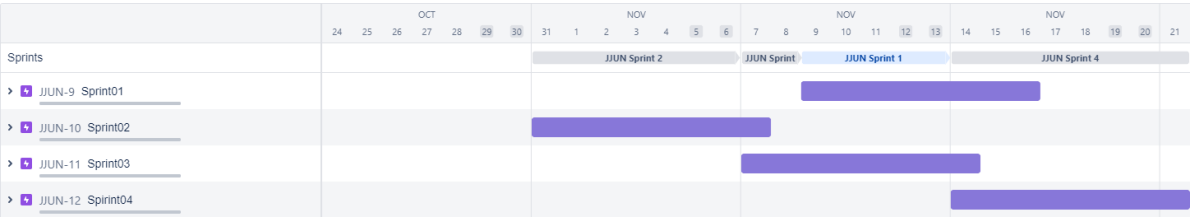
| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Team Members |
|---|---|---|---|---|---|---|
| Sprint-1 | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | 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 |

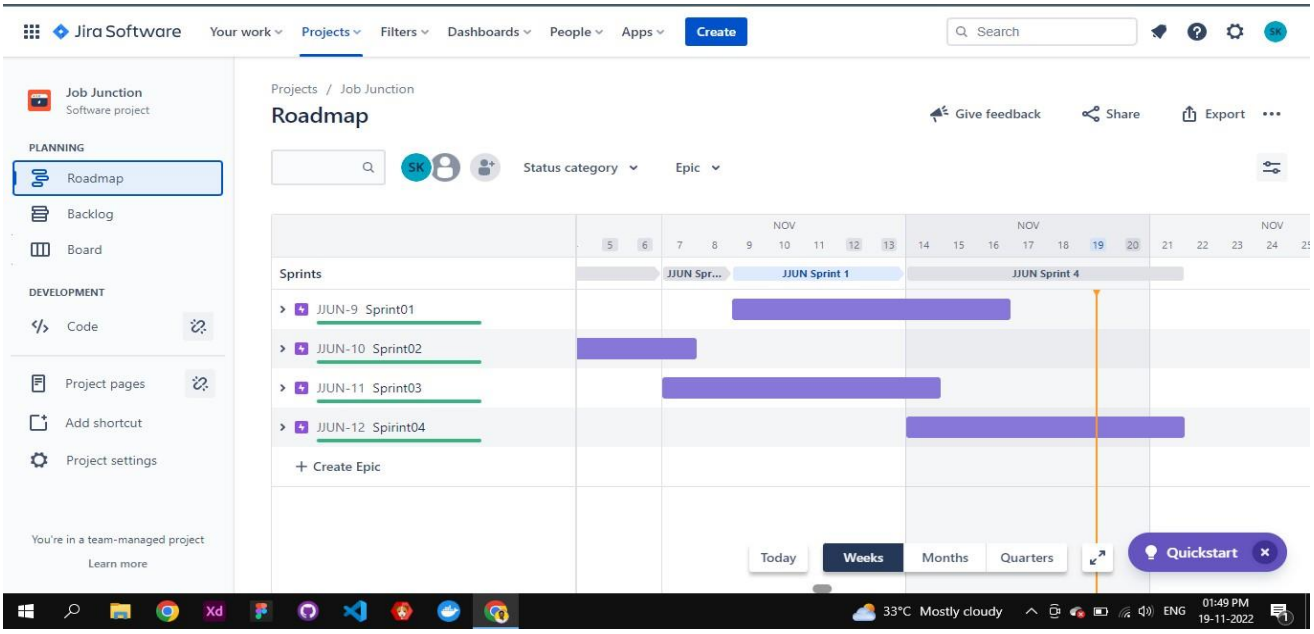
## 6.2 Sprint Delivery Schedule

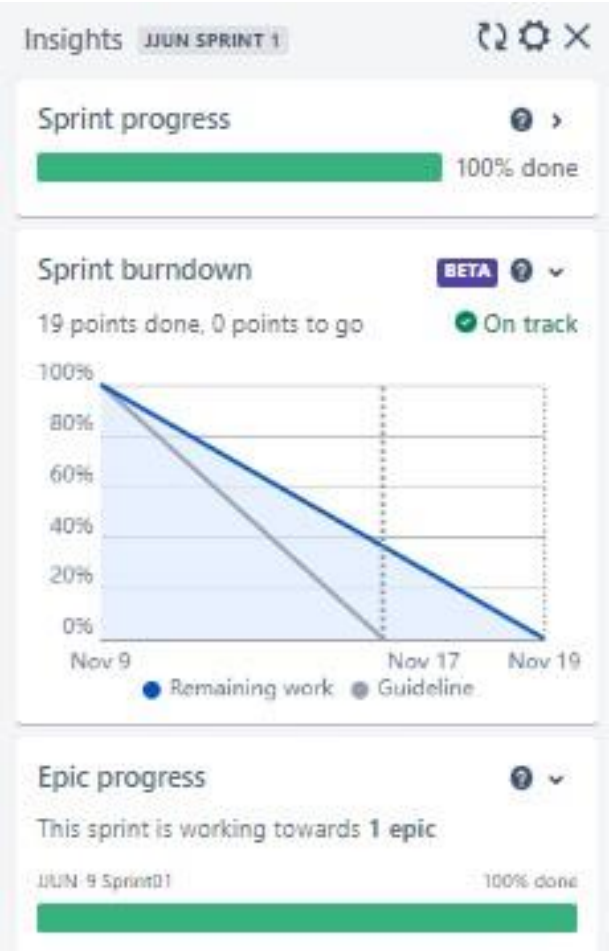
**Project Tracker, Velocity & Burndown Chart: (4 Marks)**

| 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

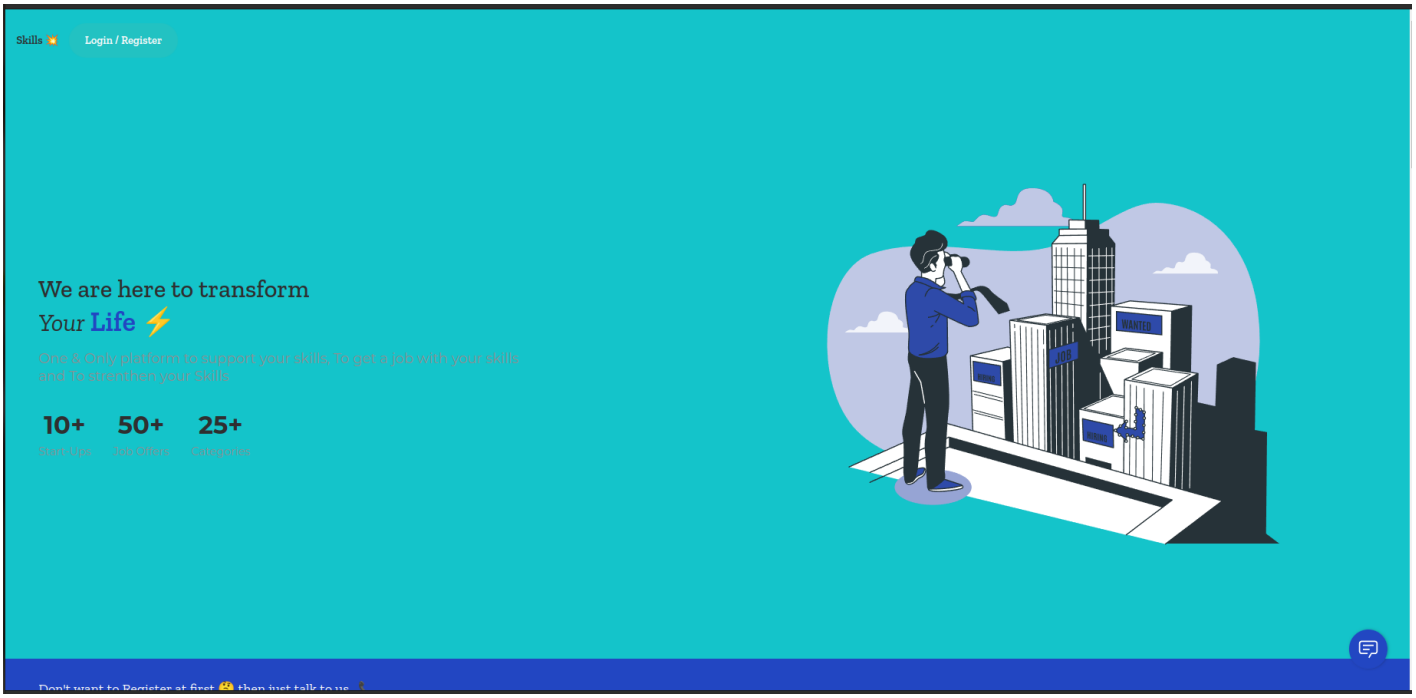



Insights
JJUN SPRINT 1
Sprint progress
100% done
Sprint burndown
BETA
19 points done, 0 points to go
On track
100%
80%
60%
40%
20%
0%
Nov 9
Nov 17
Nov 19
Remaining work
Guideline
Epic progress
This sprint is working towards 1 epic
JJUN 9 Sprint01
100% done

# 7.CODING & SOLUTIONING

## 7.1 Chatbot Integration Code



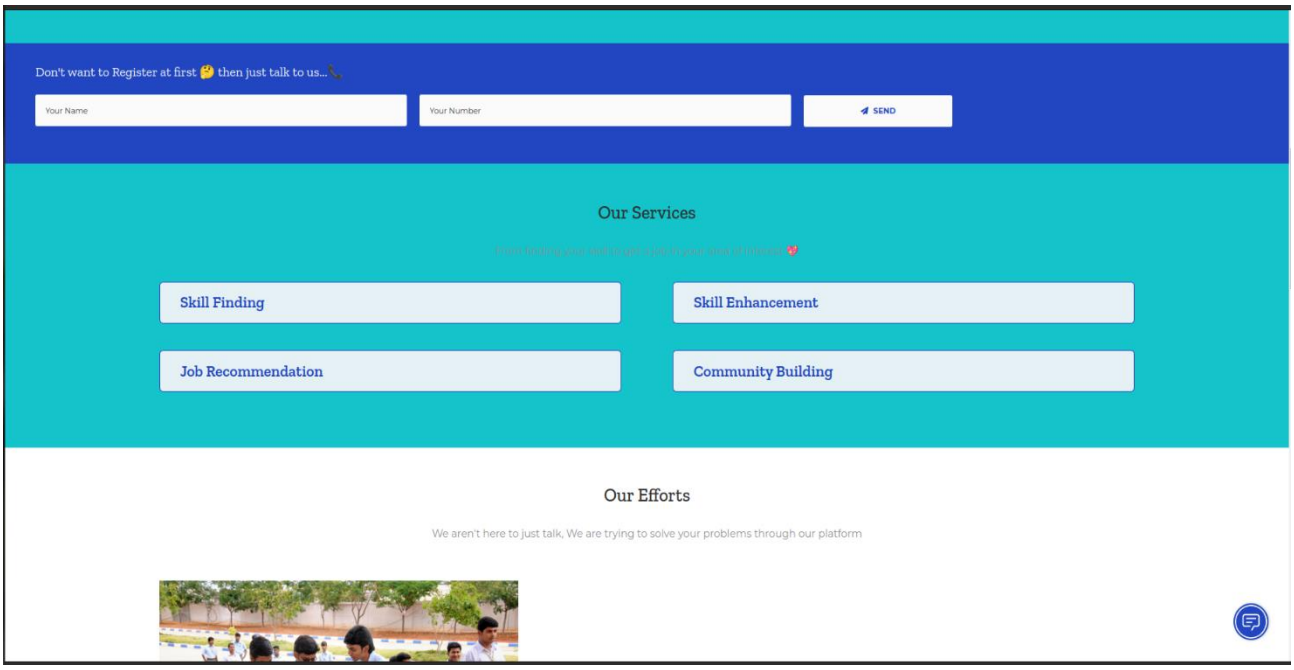

**Code**

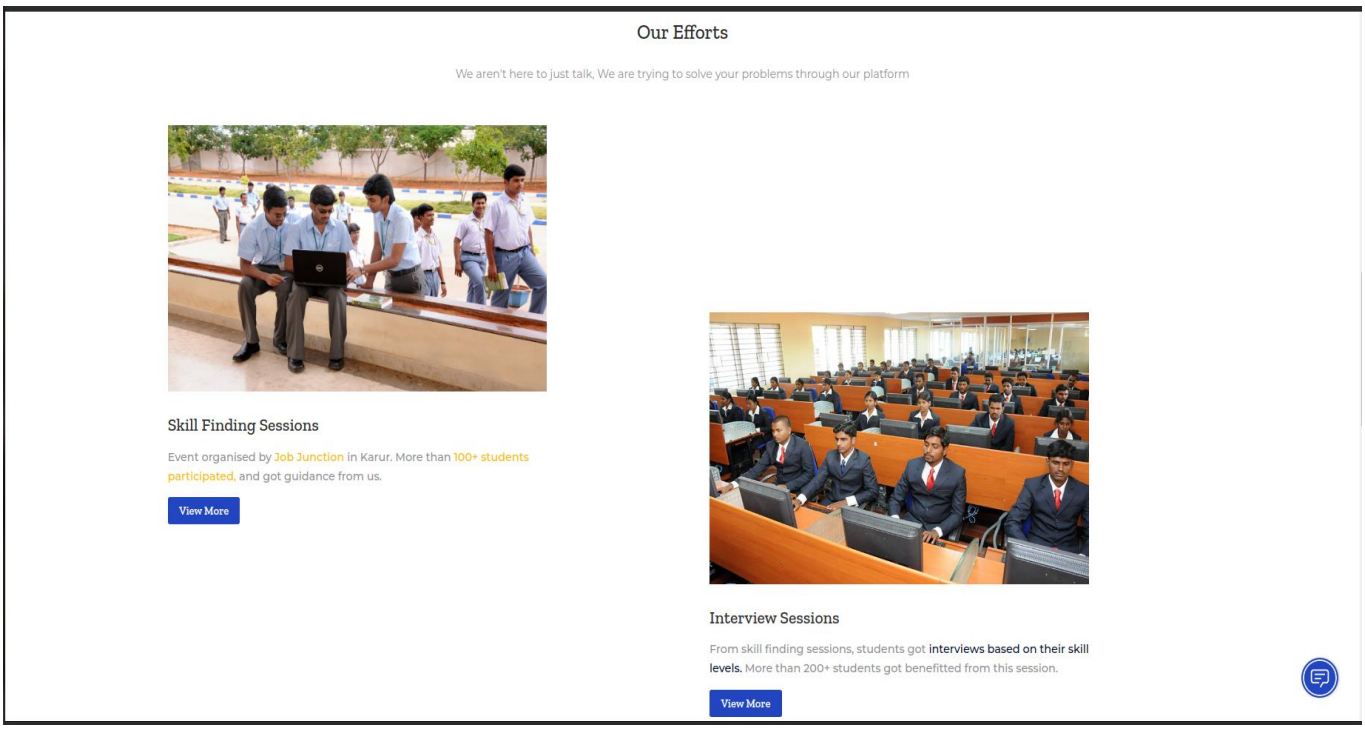
```
<script>
  window.watsonAssistantChatOptions = {
    integrationID: "c9539449-0fd4-4aca-8b8b-3a6c109a71fa", // The ID of this integration.
    region: "jp-tok", // The region your integration is hosted in.
    serviceInstanceID: "39284af4-f9b4-4159-be01-8ca534e21855", // The ID of your service
instance.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

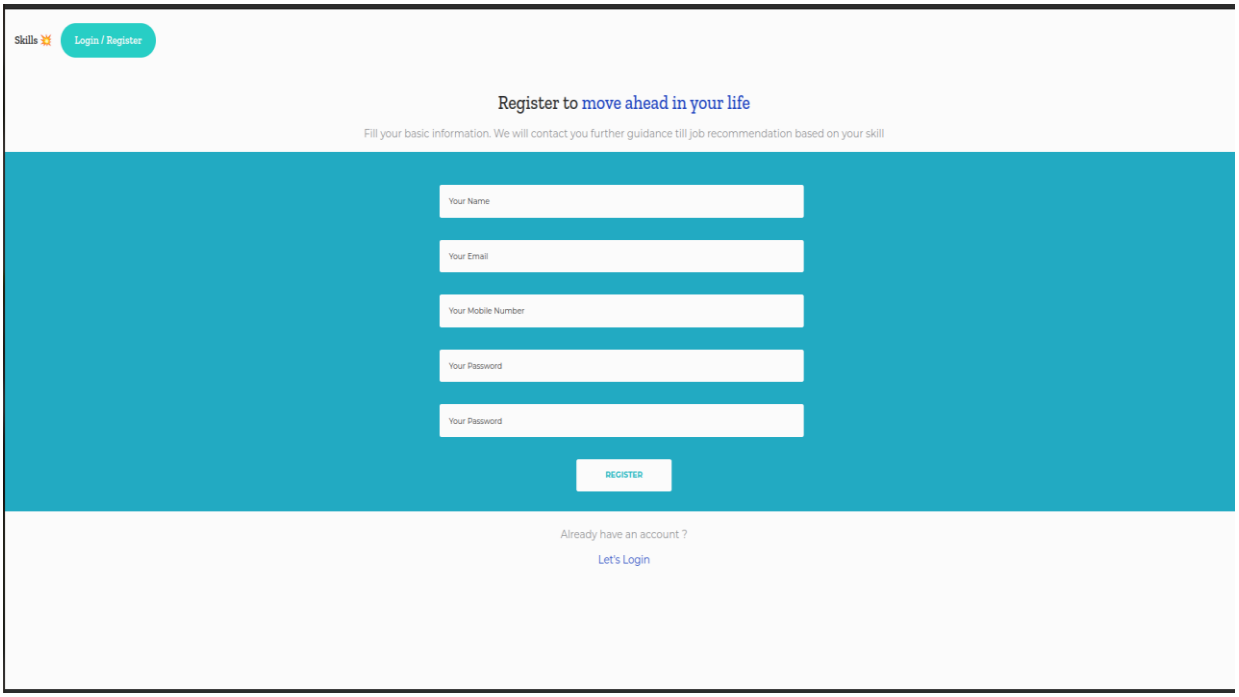
### Desktop View



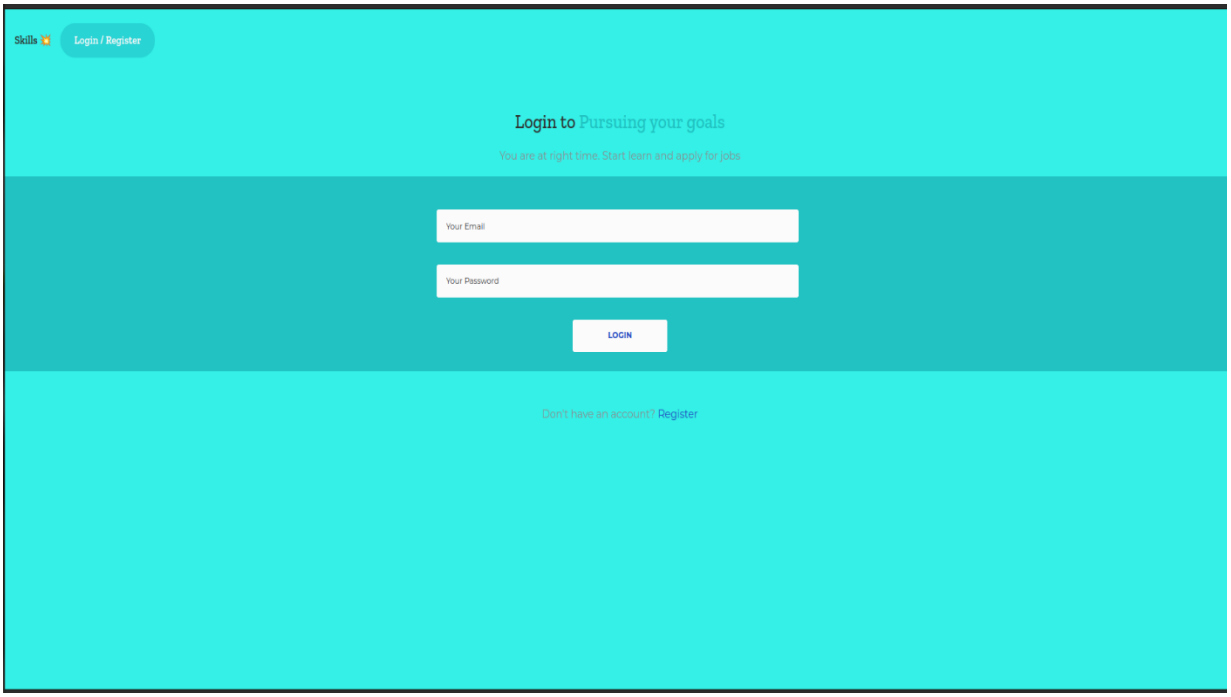



## Dynamic Application

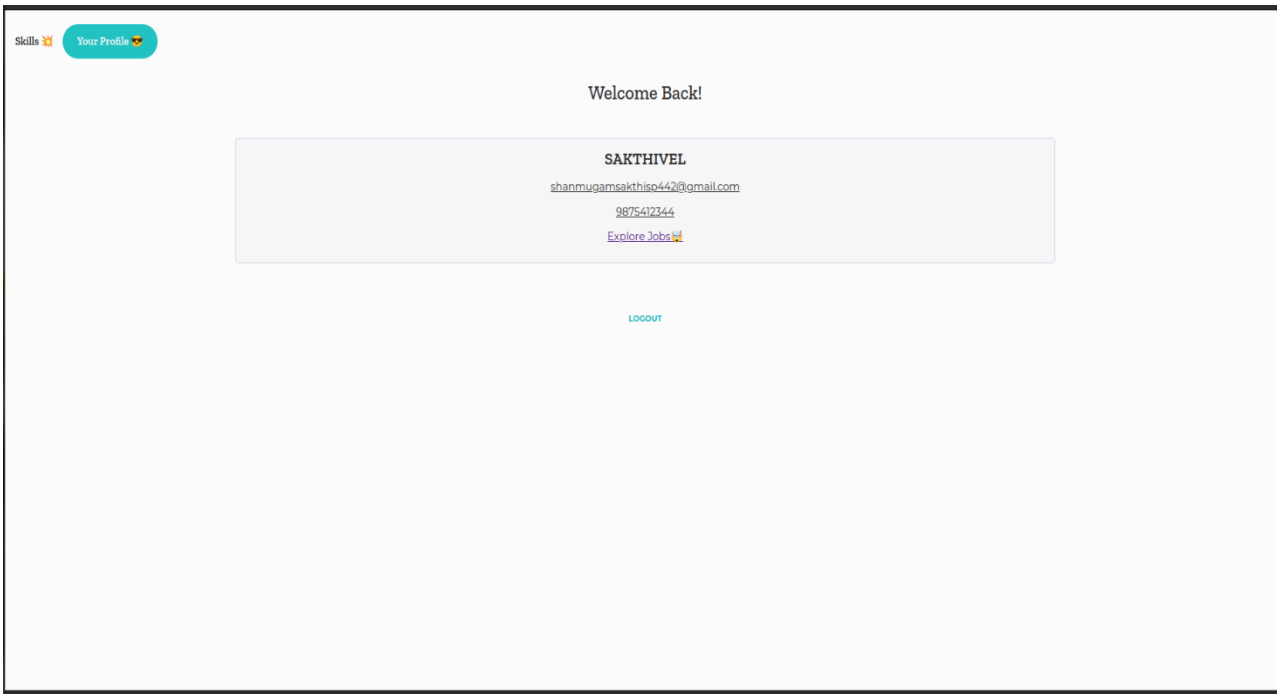
## Registration Form



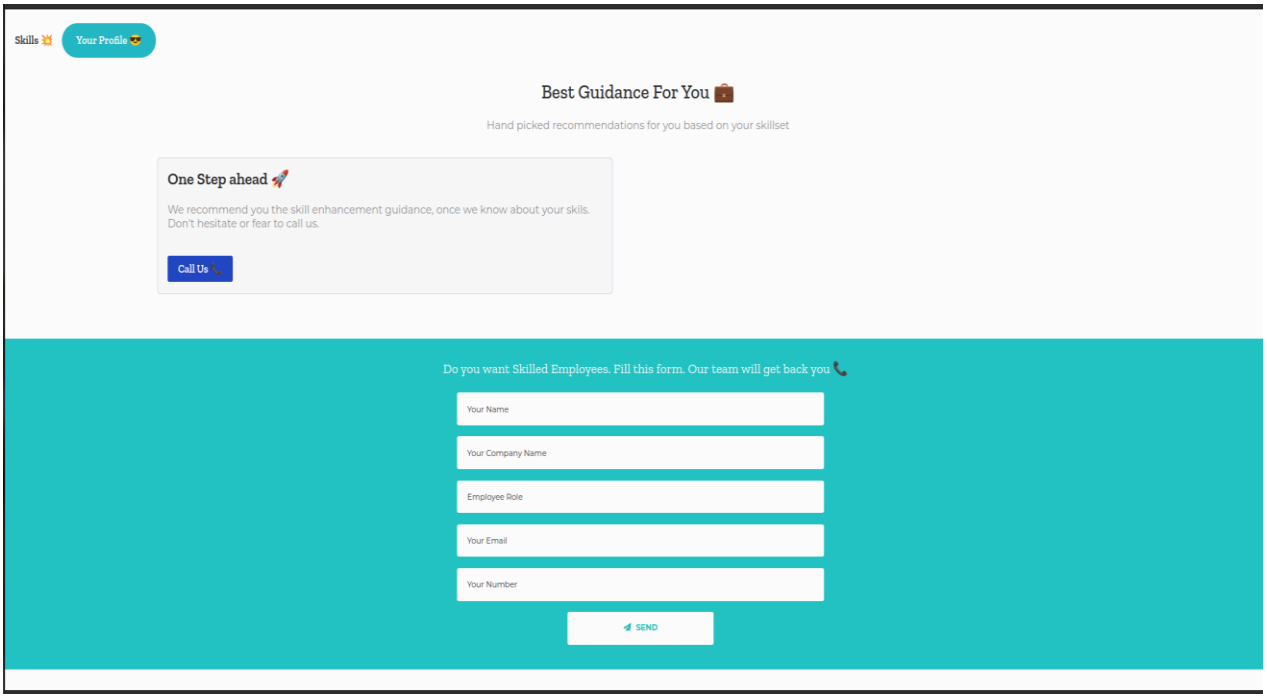

## Login After Registration



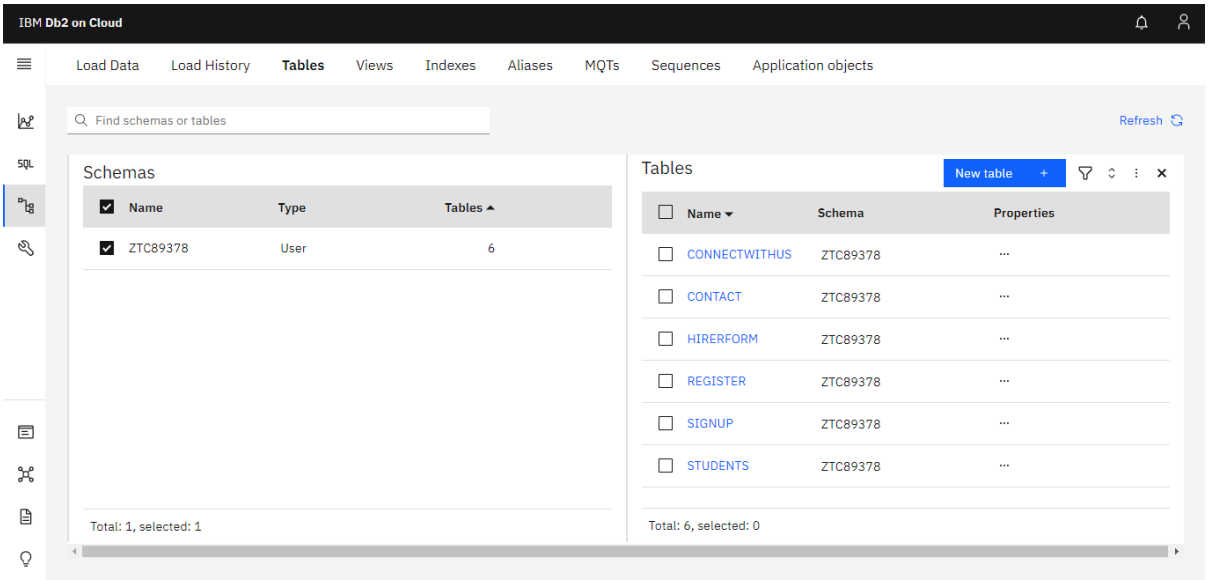
**Profile After Login**



## Jobs After Login



### 7.3 Database Schema




# 8.TESTING

## 8.1 TEST CASES

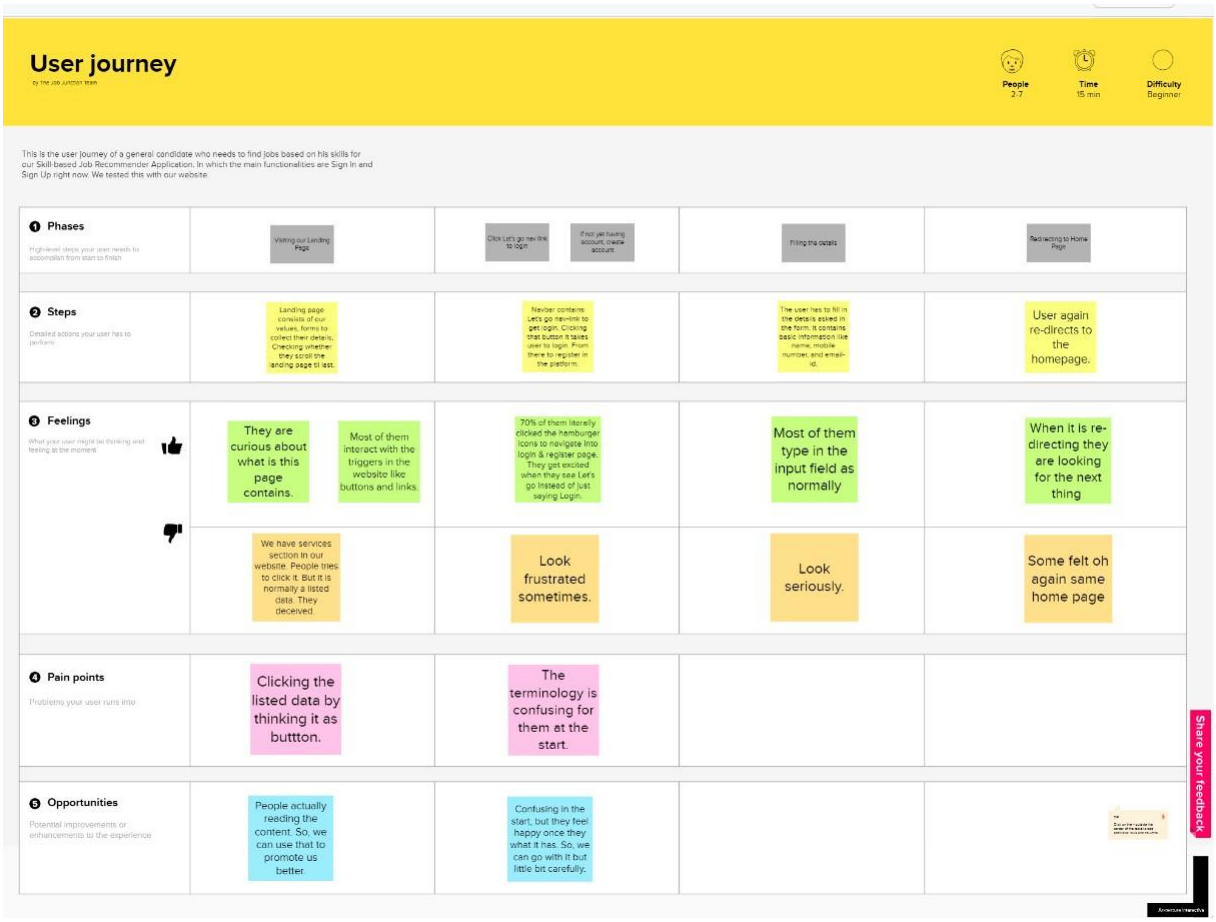
**1. Website functionality testing :** It includes Outbound links, Internal links correctness, Thereare no links leading to the same page, There are no broken links.

**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.

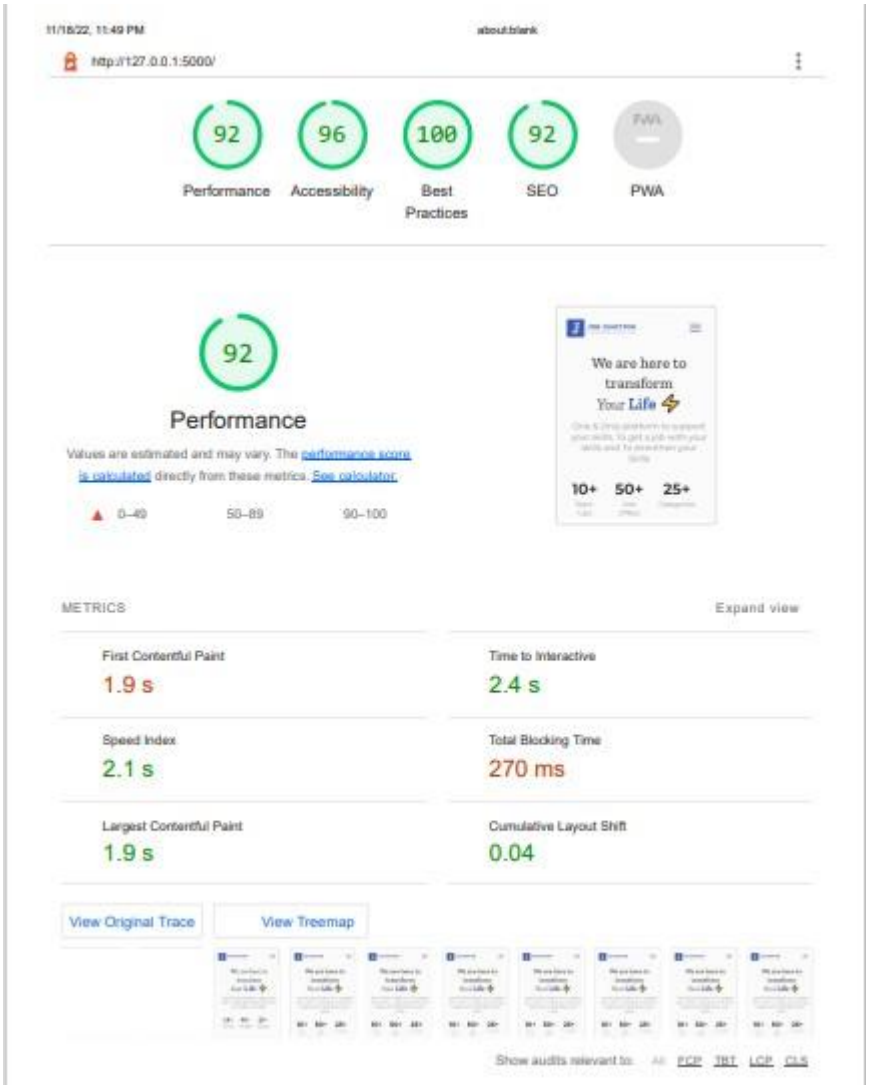
## 8.2 User Acceptance Testing




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">
            <img src="../static/images/job_junction-main.png" alt="Job
Junction Logo" />
        </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>
```

```
                <p class="hero_left-heading">We are here to transform<br><span
class="special_secondary">Your </span><span class="special_primary">Life
⚡</span></p>
                <p class="hero_left-content">One & Only platform to support
your skills, To get a job with your skills and To strenthen your Skills</p>
                <div class="hero_left-statistics">
                    <div class="hero_left-stat">
                        <p class="hero_left-statnum">10+</p>
                        <p class="hero_left-statlabel">Start-Ups</p>
                    </div>
                    <div class="hero_left-stat">
                        <p class="hero_left-statnum">50+</p>
                        <p class="hero_left-statlabel">Job Offers</p>
                    </div>
                    <div class="hero_left-stat">
                        <p class="hero_left-statnum">25+</p>
                        <p class="hero_left-statlabel">Categories</p>
                    </div>
                </div>
            </div>
        </div>
        <div class="_hero hero_right">
            <img src="{{url_for('static', filename='images/job_junction-hero-
image.svg')}}" alt="job junction hero image">
        </div>
    </div>

    <div class="header_bottom">
        <p>Don't want to Register at first 🤔 then just talk to us...📞 </p>
        {{ cmsg }}
        <form action="{{ url_for('contactrec') }}" method="POST">
        <input type="text" name = "name" placeholder="Your Name">
            <input type="tel" name = "number" placeholder="Your Number" >
            <button type="submit" name="send" class="button_send"><i
class="fa-solid fa-paper-plane icon"></i>SEND</button> <br><br>
        </form>
    </div>

    <div class="ourservices">
        <div class="header">
            <p class="head">Our Services</p>
            <p class="description">From finding your skill to get a job in
your area of interest 💖</p>
        </div>
        <div class="body">
            <div class="content">
                <div class="content_service">
                    <p>Skill Finding</p>
```

```
                </div>

                <div class="content_service">
                    <p>Skill Enhancement</p>
                </div>

                <div class="content_service">
                    <p>Job Recommendation</p>
                </div>

                <div class="content_service">
                    <p>Community Building</p>
                </div>
            </div>
        </div>
    </div>


    <div class="ourworks">
        <div class="header">
            <p class="head">Our Efforts</p>
            <p class="description">We aren't here to just talk, We are trying
to solve your problems through our platform</p>
        </div>
        <div class="body">
            <div class="content">
                <div class="content_project">
                    <img src="{{url_for('static',
filename='images/effort1.jpg')}}" alt="Job Junction Skill Finding Sessions">
                    <h5>Skill Finding Sessions</h5>
                    <p>Event organised by <span style="color: #FFB800;">Job
Junction </span> in Karur. More than <span style="color: #FFB800;"> 100+
students participated,</span> and got guidance from us.</p>
                    <button>View More</button>
                </div>
                <div class="content_project">
                    <img src="{{url_for('static',
filename='images/effort2.jpg')}}" alt="Job Junction Interview Sessions">
                    <h5>Interview Sessions</h5>
                    <p>From skill finding sessions, students got <span
style="color: #040d31;">interviews based on their skill levels.</span> More
than 200+ students got benefitted from this session.</p>
                    <button>View More</button>
                </div>
            </div>
        </div>
        <div class="bottom">
            <p>We have more to show you 😎</p>
```

```
            <a href="/login">Want to see ? <span>CLICK HERE <i class="fa-solid
fa-arrow-right icon"></i></span></a>
            <img class="decimg decimg1" src="{{url_for('static',
filename='images/jobjunction_uiux.svg')}}" alt="UIUX Design">
            <img class="decimg decimg2" src="{{url_for('static',
filename='images/jobjunction_appdevelopment.svg')}}" alt="App Development">
            <img class="decimg decimg3" src="{{url_for('static',
filename='images/jobjunction_endtoendsolutions.svg')}}" alt="End-to-End
Solutions">
            <img class="decimg decimg4" src="{{url_for('static',
filename='images/jobjunction_webdevelopment.svg')}}" alt="Web Development">
        </div>
    </div>

    <div class="happycustomers">
        <div class="header">
            <p class="head">Our <span style="color:#2246C2">Words to
you</span></p>
            <p class="description">We suffered a lot in finding jobs. So, we
are here to solve it 💪</p>
        </div>
        <div class="body">
            <div class="content">
                <div class="content_feedbacks">
                    <div class="feedback">
                        <p>In our beginning of our career as a fresher, We
struggled a lot to find jobs. Platforms like LinkedIn, Naukri and Indeed
suggest jobs just by our keywords and resume. It doesn't provide the exact
jobs for us. Company's description for aren't optimised for freshers. So, we
talk to you and recommend you the jobs, courses to take and personalised
guidance in a more human-centered way 👨‍🤝‍👨</p>
                        <h4>- Job Junction Team</h4>
                    </div>
                    <img class="quote" src="{{url_for('static',
filename='images/quote.svg')}}" alt="Job Junction Customers Feedback Quote">
                </div>
                <div class="content_svg">
                    <img src="{{url_for('static',
filename='images/jobjunction_customer_svg.svg')}}" alt="Job Junction Happy
Customers.">
                </div>
            </div>
        </div>
    </div>


    <div class="footer">
        <div class="header">
```

```
            <p class="head">Want to <span style="color:#2246C2">Connect With
Us</span></p>
            <p class="description">Don't hesitate. We are here to clear your
confusions. Feel Free to contact Us.</p>
        </div>
        <div class="body">
        <div class="content">
            {{ cnmsg }}
                <form action="{{ url_for('connectrec') }}" method="POST"
class="form_footer">
                    <div>
                        <input type="text" name="CustomerName"
placeholder="Your Name" >
                        <input type="email" name="MailID" placeholder="Your
Email" >
                        <input type="tel" name="ContactNumber"
placeholder="Your Mobile Number" >
                        <input type="time" name="ConvenientTime"
placeholder="Your Convinient time to speak" >
                        <input type="text" name="CustomerMessage"
placeholder="Your Message in your language">
                    </div>
                    <div><button type="submit" name="connect"
class="button_send" style="margin:16px auto;"><i class="fa-solid fa-paper-
plane icon"></i>CONNECT</button></div>
                </form>
            </div>
        </div>
        <div class="bottom">
            <div class="line"></div>
            <div class="header">
                <p class="head">More Ways To Reach Us</p>
                <p
onclick="location.href='https://in.linkedin.com/in/sunilprasadkuppusamy';"
class="head" style="color: lightblue; margin-top:16px; margin-bottom:32px;
cursor: pointer; line-height: normal; font-family: 'Montserrat'; font-style:
normal;font-weight: 400;">With 🤍 from Job Junction</p>
            </div>
            <div class="contact_info" style="cursor:pointer;">
                <p
onclick="location.href='mailto:sunilprasadk2001@gmail.com'"><span><i
class="fa-solid fa-envelope"></i></span> contact@job-junction.com</p>
                <p onclick="location.href='tel:9080244817'"><span><i
class="fa-solid fa-phone"></i></span> +91 9080244817</p>
            </div>
            <div class="social_info">
                <div style="text-align: center;">
                    <a href="#">FACEBOOK</a>
```

```
                    <span><i class="fa-solid fa-circle" style="color: #D9D9D9;
font-size: 8px; margin:auto 16px;"></i></span>
                    <a href="#">TWITTER</a>
                    <span><i class="fa-solid fa-circle" style="color: #D9D9D9;
font-size: 8px; margin:auto 16px;"></i></span>
                    <a href="#">LINKEDIN</a>
                </div>
            </div>
            <div class="footer_bottom">
                <p>2022 Job Junction Pvt Ltd | All rights reserved </p>
            </div>
        </div>
    </div>
</body>
```

**#register.html**

## ----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;
```

```
}

/*ourgrowths - body - content container*/
.ourgrowths .body .growths{
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
}

/*ourgrowths - body - growth container*/
.growth{
    align-items: center;
    background: #f6f6f6;
    border: 1px solid #d5daed;
    border-radius: 6px;
    padding:16px 16px;
    margin-right:2.5%;
    margin-left:2.5%;
    margin: 20px 16px;
}

/*ourgrowths - body - growth container - p*/
.growth p{
    font-family: 'Zilla Slab';
    font-style: normal;
    font-weight: 600;
    font-size: 20px;
    color: #2F2F30;
}

/*ourgrowths - body - growth container - title*/
.growth .growth_title{
    color:#2F2F30;
}

/*ourgrowths - body - growth container - category*/
.growth .growth_category{
    display: inline-block;
    background-color:#395bc1;
    color:white;
    padding:5px 10px;
    border-radius: 25px;
    font-size:16px;
    margin-top: 8px;
}


@media only screen and (max-width: 1040px){
    .growth p{
```

**---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;SECURIT
Y=SSL;SSLServerCertificate=DigiCertGlobalRootCA.crt;UID=ztc89378;PWD=HSYKGIqGB
cRZVRHK",'','')
print("Connected Successfully !")


app = Flask(_name_)

app.secret_key = 'Done ehhh'


@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(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, number)
            ibm_db.bind_param(prep_stmt, 3, email)
            ibm_db.bind_param(prep_stmt, 4, password)
            ibm_db.bind_param(prep_stmt, 5, cpassword)
            ibm_db.execute(prep_stmt)

        return render_template('login.html', msg="Registered
successfuly..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(prep_stmt, 1, Name)
        ibm_db.bind_param(prep_stmt, 2, MobileNumber)
        ibm_db.execute(prep_stmt)

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

```python
@app.route("/connectrec",methods = ['POST', 'GET'])
def connectrec():
    cnmsg=''
    if request.method == 'POST':

        CustomerName = request.form['CustomerName']
        MailID = request.form['MailID']
        ContactNumber = request.form['ContactNumber']
        ConvenientTime = request.form['ConvenientTime']
        CustomerMessage = request.form['CustomerMessage']
        insert_sql = "INSERT INTO connectwithus VALUES (?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, CustomerName)
        ibm_db.bind_param(prep_stmt, 2, MailID)
        ibm_db.bind_param(prep_stmt, 3, ContactNumber)
        ibm_db.bind_param(prep_stmt, 4, ConvenientTime)
        ibm_db.bind_param(prep_stmt, 5, CustomerMessage)
        ibm_db.execute(prep_stmt)

        return render_template('index.html', cnmsg="Lets Travel Together")


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

        CandidateName = request.form['CandidateName']
        CompanyName = request.form['CompanyName']
        EmployeeRole = request.form['EmployeeRole']
        CandidateMail = request.form['CandidateMail']
        CandidateNumber = request.form['CandidateNumber']
        insert_sql = "INSERT INTO hirerform VALUES (?,?,?,?,?)"
        prep_stmt = ibm_db.prepare(conn, insert_sql)
        ibm_db.bind_param(prep_stmt, 1, CandidateName)
        ibm_db.bind_param(prep_stmt, 2, CompanyName)
        ibm_db.bind_param(prep_stmt, 3, EmployeeRole)
        ibm_db.bind_param(prep_stmt, 4, CandidateMail)
        ibm_db.bind_param(prep_stmt, 5, CandidateNumber)
        ibm_db.execute(prep_stmt)

        return render_template('jobs.html', jmsg="We will contact you soon")



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

```
@app.route('/exploreJobs', methods = ['GET', 'POST'])
def exploreJobs():
    url = "https://linkedin-jobs-search.p.rapidapi.com/"

    payload = {
        "search_terms": "python programmer",
        "location": "30301",
        "page": "1"
    }
    headers = {
        "content-type": "application/json",
        "X-RapidAPI-Key":
"6ac028283fmshc8ee5a710f1e3c4p196a21jsnb05b95329620",
        "X-RapidAPI-Host": "linkedin-jobs-search.p.rapidapi.com"
    }

    response = requests.request("POST", url, json=payload, headers=headers)
    #print(response.text)
    job = json.loads(response.content)
    #print(job)
    #json_data = json.loads(job)
    return render_template('exploreJobs.html', jobs =job)


@app.route('/logout')
def logout():
    # Remove session data, this will log the user out
   session.pop('email', None)
   session.pop('password', None)
   session.pop('name', None)
   session.pop('number',  None)
   # Redirect to login page
   return redirect(url_for('login'))

if_name_== "_main_":
    app.run(debug=True)
```

**#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")
```

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