# SKILLS AND JOB RECOMMENDER APPLICATION

**IBM-Project-22849-1659859395**

## NALAIYA THIRAN PROJECT BASED LEARNING ON PROFESSIONAL READLINESS FOR INNOVATION, EMPLOYNMENT AND ENTERPRENEURSHIP

**PROJECT REPORT**

**ROSHAN S(211719106065)**

**SAICHAND R(211719106068)**

**SAIKRISHNA BALAJI(211719106069)**

**VIGENSH V(211719106095)**

**BACHELOR OF TECHNOLOGY IN INFORMATION TECHNOLOGY**

**RAJALAKSHMI INSTITUTE OF TECHNOLOGY Chennai-600 124**

## INDEX

### 1. INTRODUCTION

1. Project Overview
2. Purpose

#### 2. LITERATURE SURVEY

1. Existing problem
2. Problem Statement Definition

#### 3. IDEATION & PROPOSED SOLUTION

1. Empathy Map Canvas
2. Ideation & Brainstorming
3. Proposed Solution
4. Problem Solution fit

#### 4. REQUIREMENT ANALYSIS

1. Functional requirement
2. Non-Functional requirements

#### 5. PROJECT DESIGN

1. Data Flow Diagrams
2. Solution & Technical Architecture
3. User Stories

#### 6. PROJECT PLANNING & SCHEDULING

1. Sprint Planning & Estimation
2. Sprint Delivery Schedule
3. Reports from JIRA

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

1. Feature 1
2. Feature 2
3. **TESTING**

1. User Acceptance Testing

1. **ADVANTAGES & DISADVANTAGES**
2. **CONCLUSION**

#### 11. FUTURE SCOPE

**132 APPENDIX**

Source Code GitHub

## 1. INTRODUCTION

### 1.1 PROJECT OVERVIEW

Category: Cloud App Development

Team ID: PNT2022TMID27168

Skills Required: IBM Cloud, HTML, JavaScript, IBM Cloud Object Storage, Python Flask , Kubernetes, Docker ,IBM DB2,IBM Container Registry

The Skills and job Recommender application is an online application that ensures there is proper representation for every skill and job related to them. It is aimed at making the process of finding job much easier by providing a resolute and clear perspective of the wide variety of job opportunities available in variety of field of skillsets. The main aim being to create an application that will serve the purpose of helping aspiring job applicants and present to them the most appropriate and suitable of job opportunities matching their skillsets.

### 1.2 PURPOSE

This project is aimed to developing an online job recommender application. The entire project has been developed keeping in view of the distributed client server computing technology in mind.

The Skills and job Recommender application is an online application that ensures there is proper representation for every skill and job related to them. It is aimed at making the process of finding job much easier by providing a resolute and clear perspective of the wide variety of job opportunities available in variety of field of skillsets. The main aim being to create an application that will serve the purpose of helping aspiring job applicants and present to them the most appropriate and suitable of job opportunities matching their skillsets

The project has been planned to be having the view of distributed architecture" with centralized storage of the database. The application for the storage of the data has been planned. Using the constructs of IBM database services and all the user interfaces have been designed using the ASP.Net technologies.

The database connectivity is planned using the "SQL Connection" methodology. The standards of security and data protective mechanism have been kept in consideration and proper security measures have been included like securing API key and having a properly issued SSL certificate. The application takes care of different modules and their associated reports" which are produced as per the applicable strategies and standards that are put forwarded by the administrative staff.

The database connectivity was planned using the latest "SQL Connection" technology provided by Microsoft corporation. The authentication and authorization was cross checked at all the relevant stages. The user level accessibility has been restricted into two zones namely.

### 2. LITERATURE SURVEY

**2.1 EXISTING PROBLEM**

#### Introduction

Existing job suggestion structures only reflect inconsideration on the user's subject of pastime and omit the user's profile and skills, which should result in extra relevant career guidelines for users. CaPaR, a Career Path Recommendation framework, is proposed in this paper to address such shortcomings. The gadget scans the user's profile and resume, identifies the candidate's key skills, and generates customized job recommendations using textual content mining and collaborative filtering techniques. Furthermore, the device suggests to student's extra competencies needed for related job openings, as nicely as learning assets for each skill. As a result, the gadget not only permits its customers to explore big quantities of information, but additionally to enlarge their portfolio and resume in order to enhance their careers.

Given the quantity of profession role statistics of individuals handy online, personalised profession route recommendation systems that should mine and advocate the most relevant profession paths for a consumer are on the rise. However, such advice systems usually are solely positive inside a single company the place there are standardized job roles. At an enterprise area level such as Information Technology or across such one of a kind enterprise sectors (such as retail, insurance, health care), mining and recommending the most applicable career paths for a user is still an unsolved lookup challenge. Towards addressing this problem, this paper proposes a machine that leverages the concept of competencies to construct talent graphs that can shape the foundation for profession path recommendations. Skills are perceived to be greater amenable for profession path standardizations across the organizations. The proposed device ingests a user's profile (in a pdf, phrase format or different public and shared data sources) and leverages an Open IE pipeline to extract education and experiences. Subsequently, the extracted entities are mapped as precise capabilities that are expressed in the form of a novel unified skill graph. Such ability graphs which capture both spatial and temporal relationships are believed to aid in producing specific profession path recommendations. An comparison of this modern ability extraction mannequin with an industrial scale dataset yielded a precision and recall of 80.54% and 86.44% respectively.

#### 2.2 PROBLEM STATEMENT DEFINITION

Freshly graduated students and employee’s Looking for job change are finding it difficult to find a job suitable to their skills. Finding the right kind of job vacancies which suits your skill set perfectly is difficult. An undergraduate pass out is well versed in the field of cybersecurity and network connections finds it difficult to scout an opportunity that is suitable to their skillset. A platform that will keep her updated on job vacancies regularly and update the status of her current application

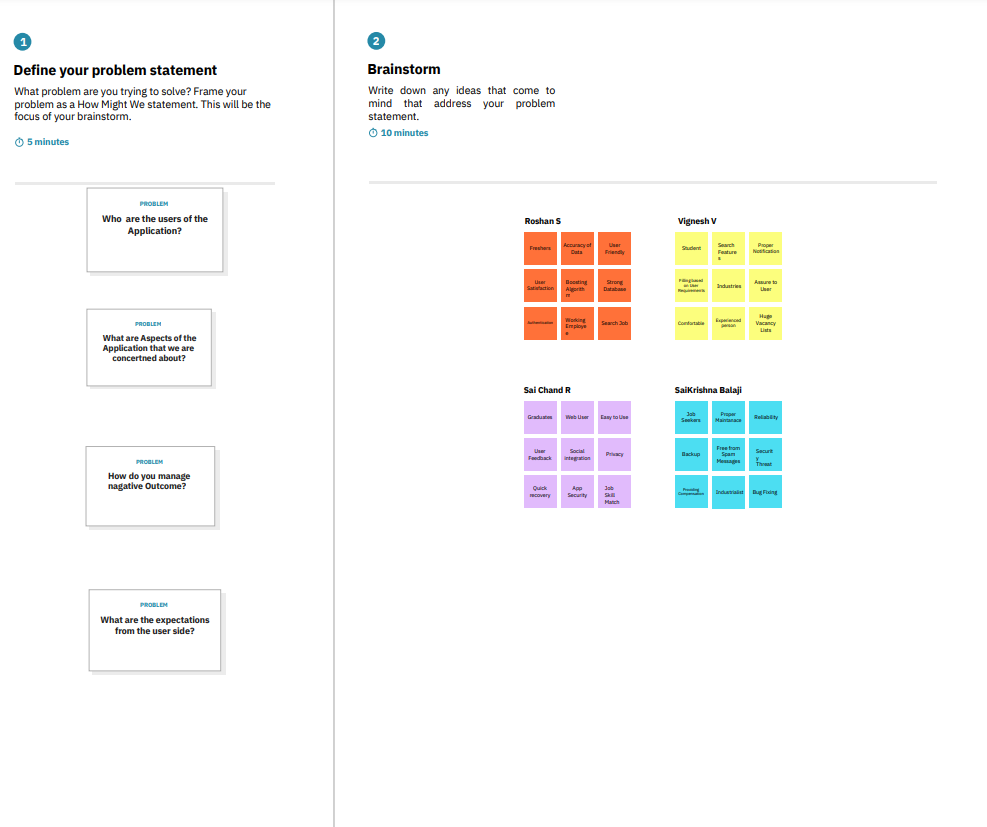
### 3. IDEATION & PROPOSED SOLUTION

#### 3.1 EMPATHY MAP CANVAS

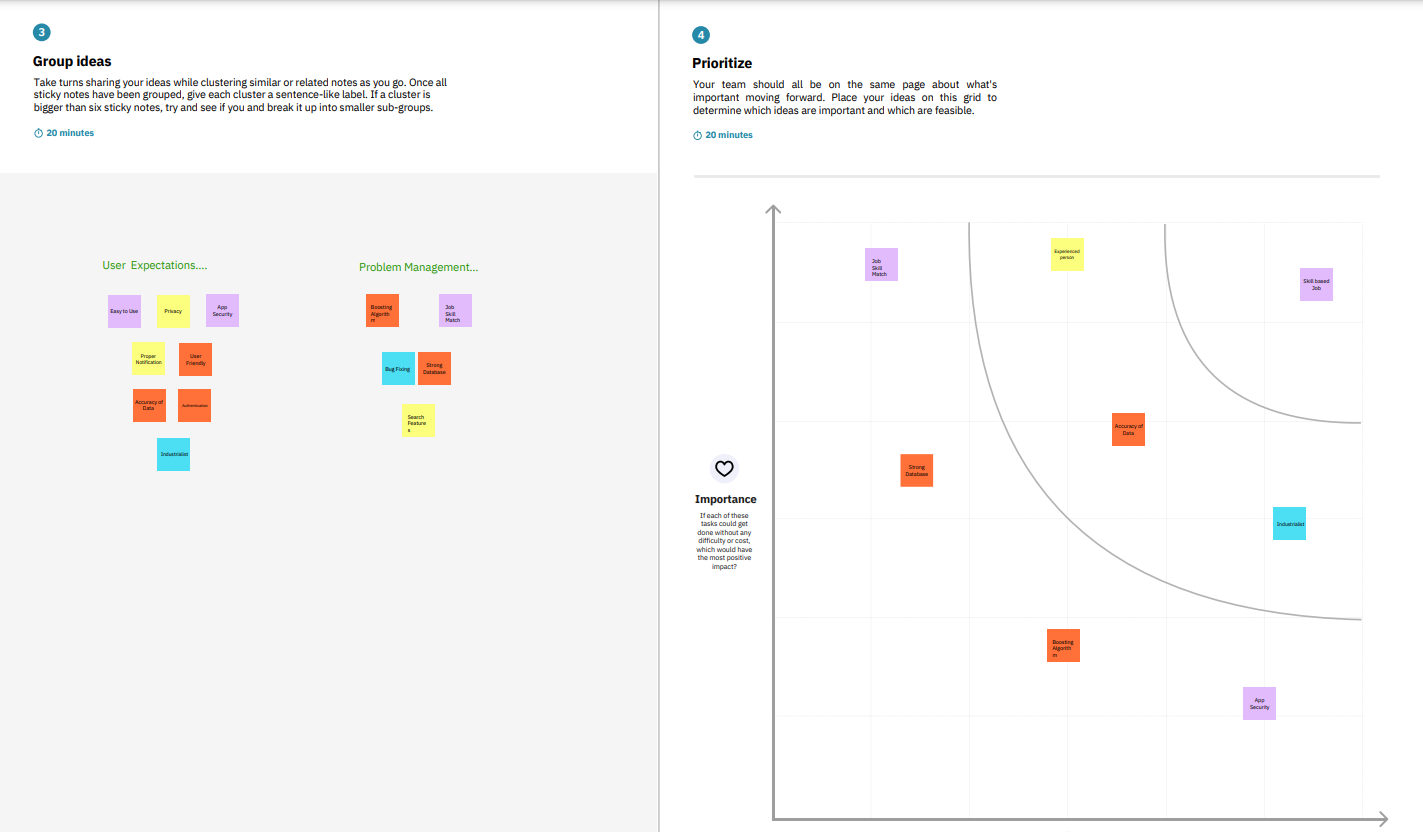


### 3.2 IDEATION & BRAINSTORMING

Step 1: Problem Statement and Brainstorm

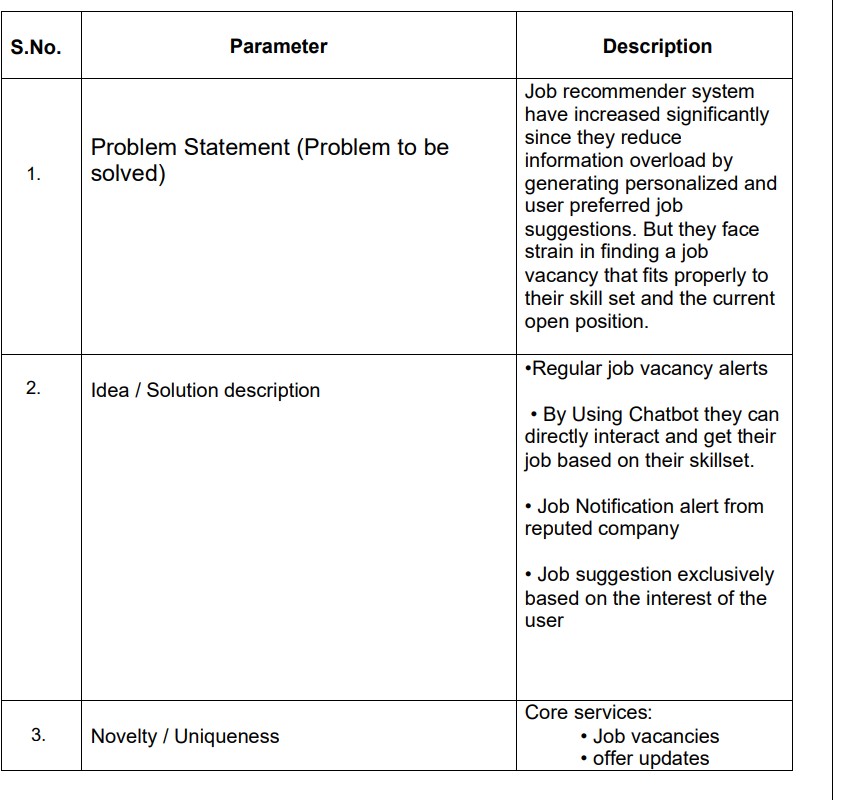


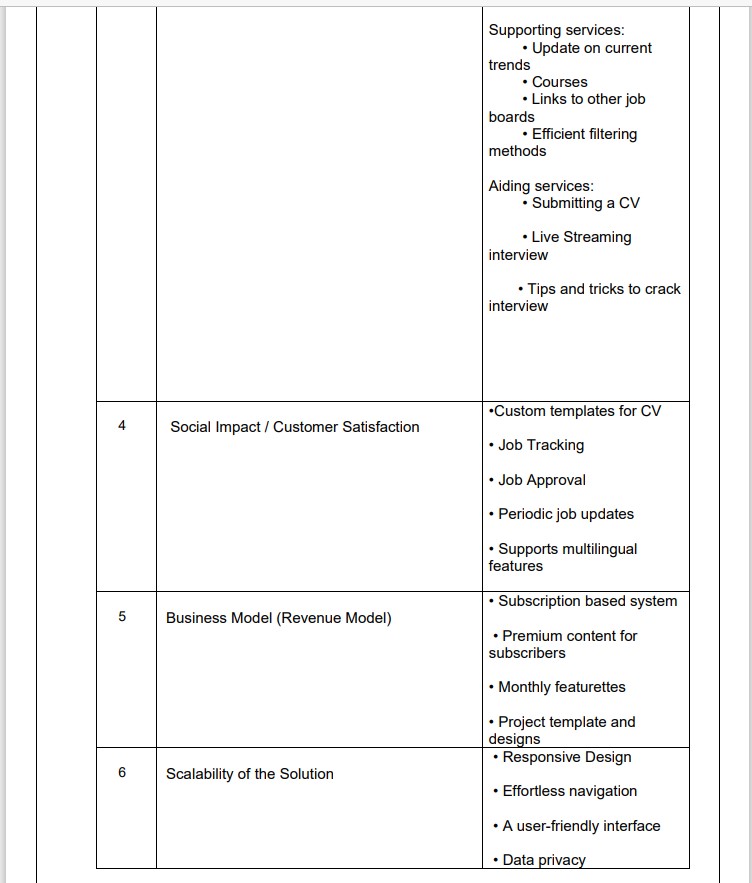
Step 2: Grouping and Idea Prioritization



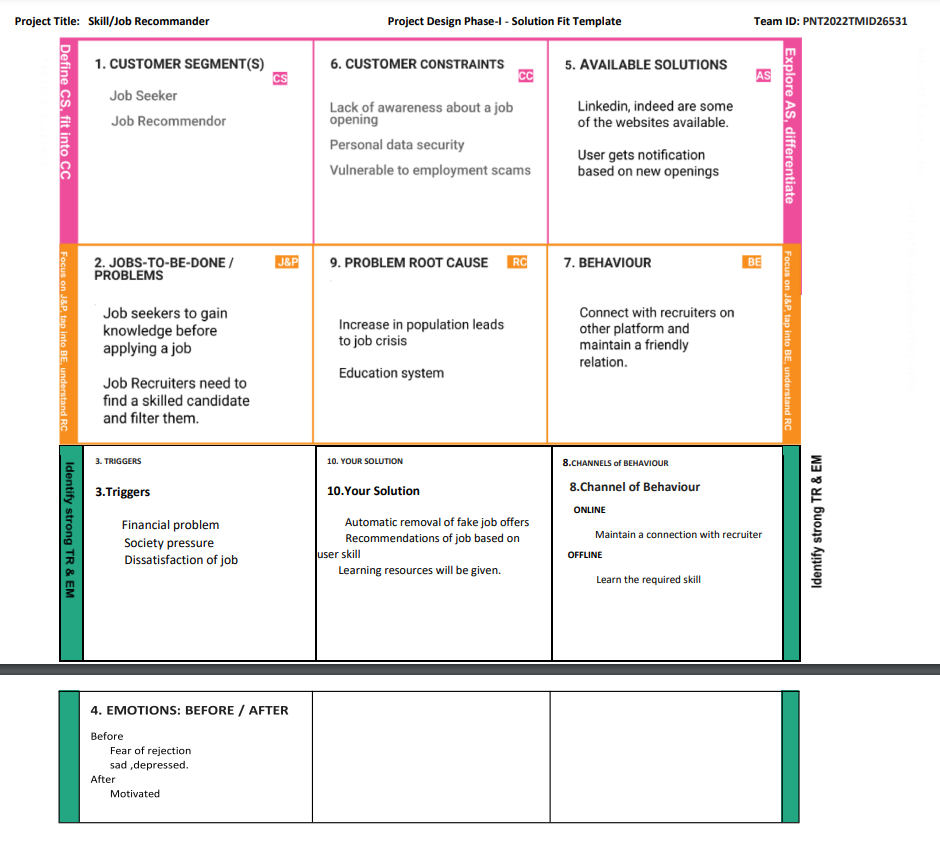
3.3 PROPOSED SOLUTION

Freshly graduated students and employee’s Looking for job change are finding it difficult to find a job suitable to their skills. Finding the right kind of job vacancies which suits your skill set perfectly is difficult. An undergraduate pass out is well versed in the field of cybersecurity and network connections finds it difficult to scout an opportunity that is suitable to their skillset. A platform that will keep her updated on job vacancies regularly and update the status of her current application



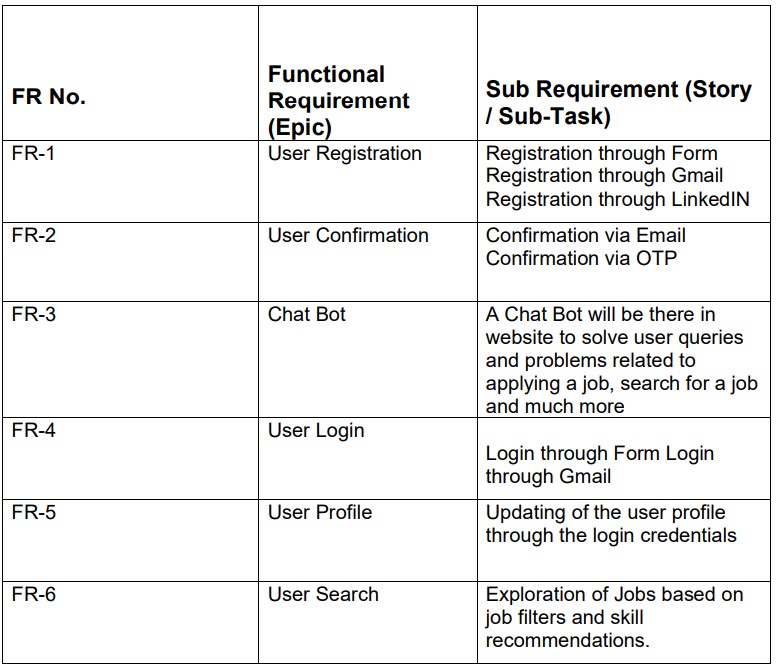


## 3.4 PROBLEM SOLUTION FIT

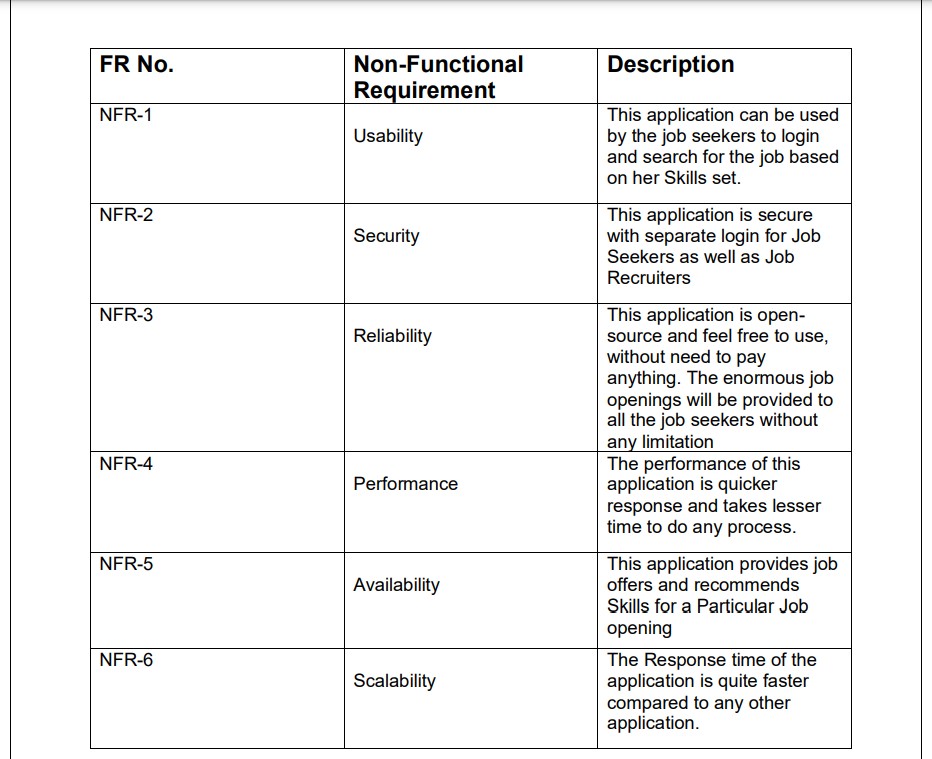
****

# 4. REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREENTS



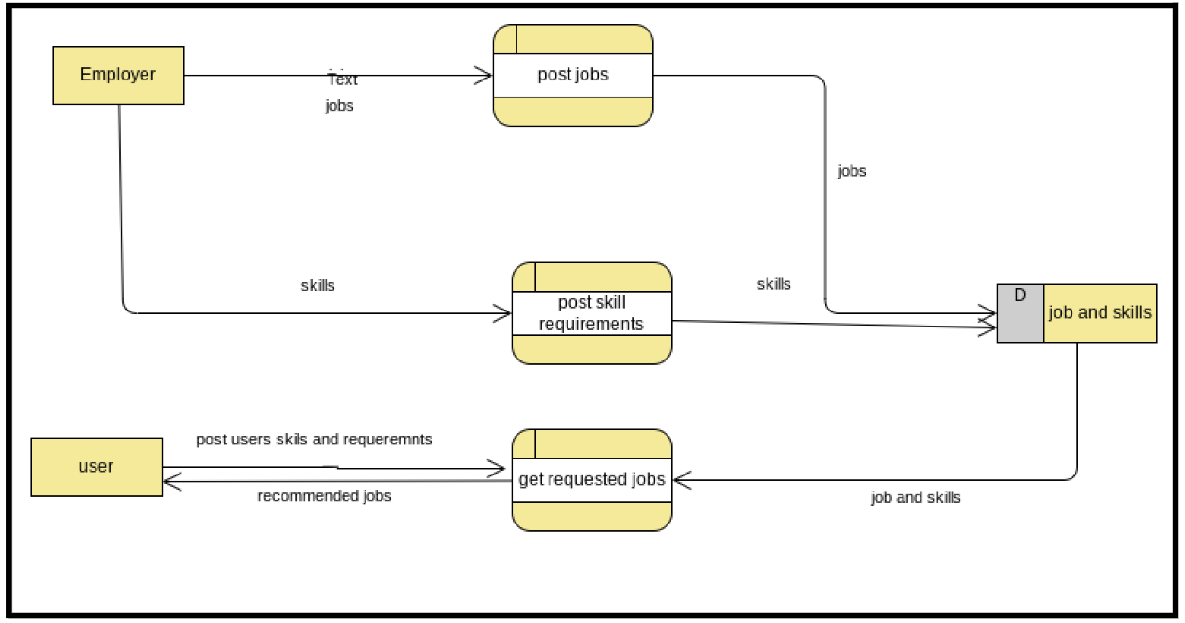
## 4.2 NON-FUNCTIONAL REQUIRMENTS



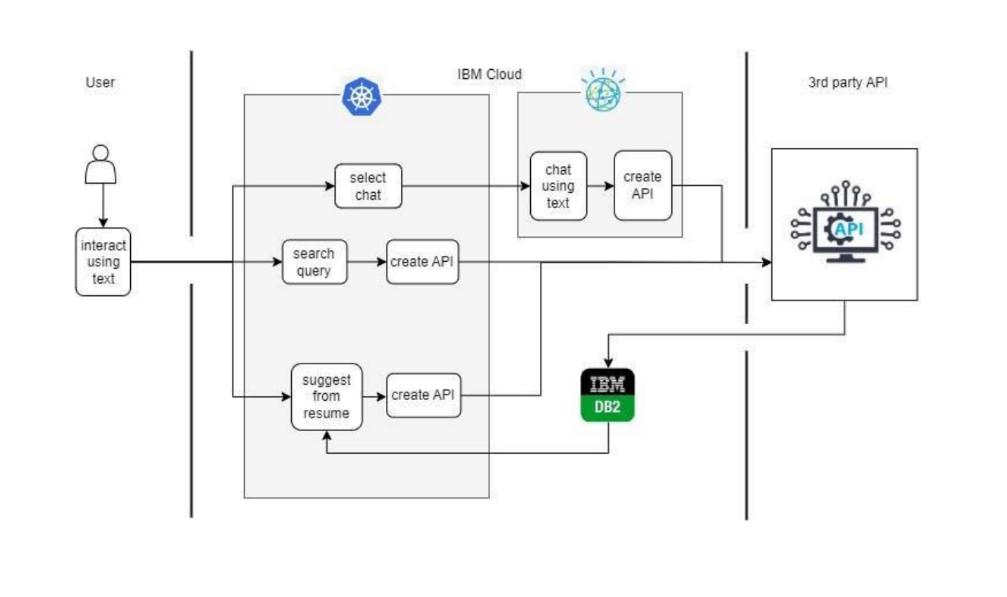
# 5. PROJECT DESIGN

## 5.1 DATA FLOW DIAGRAMS

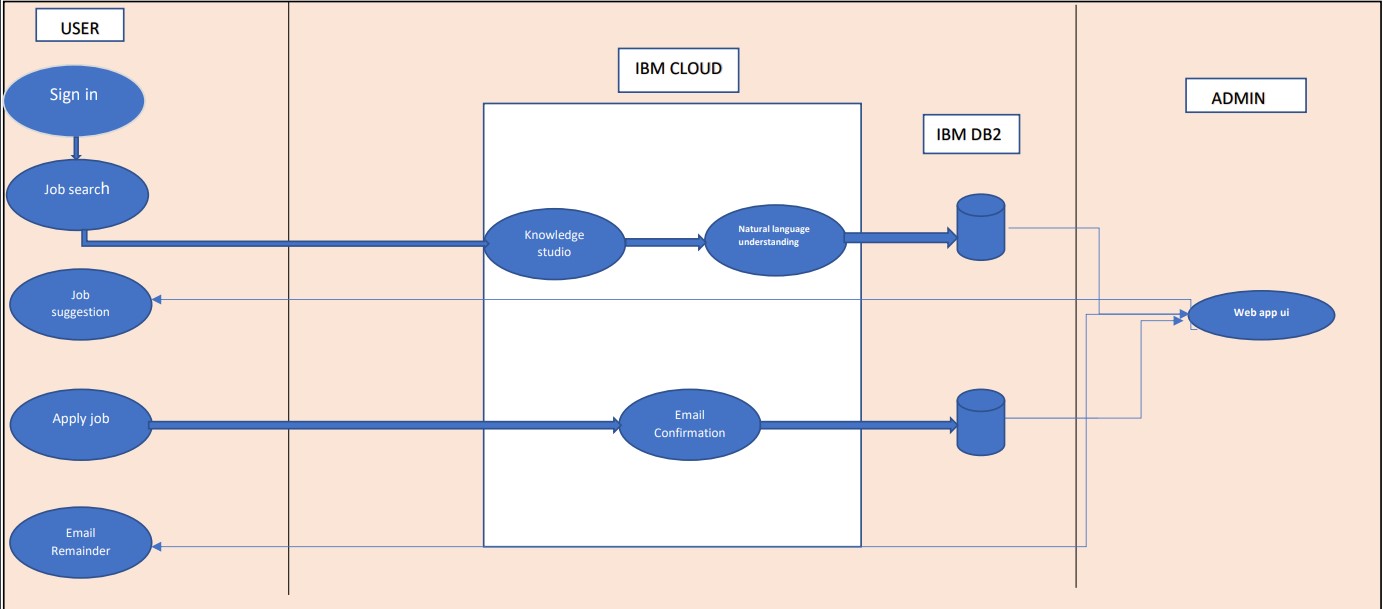
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 SOLUTION & TECHNICAL ARCHITECTURE

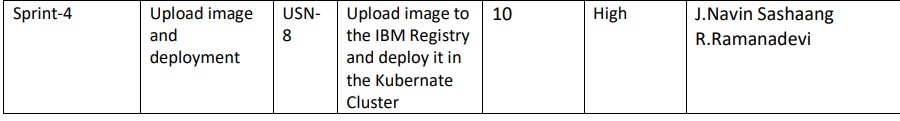
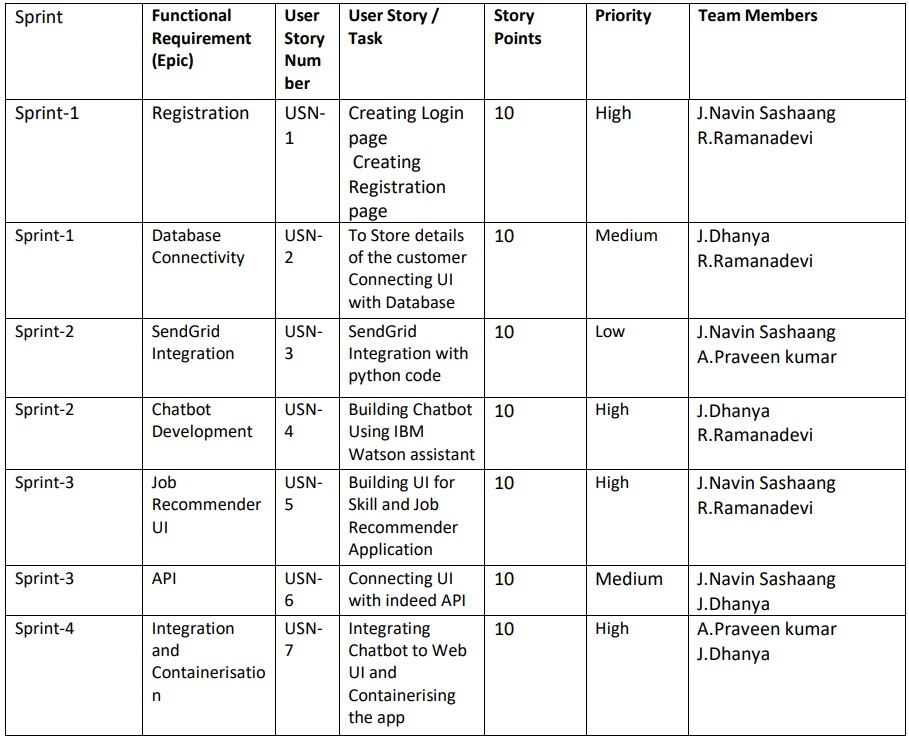
****

# SOLUTION ARCHITECTURE

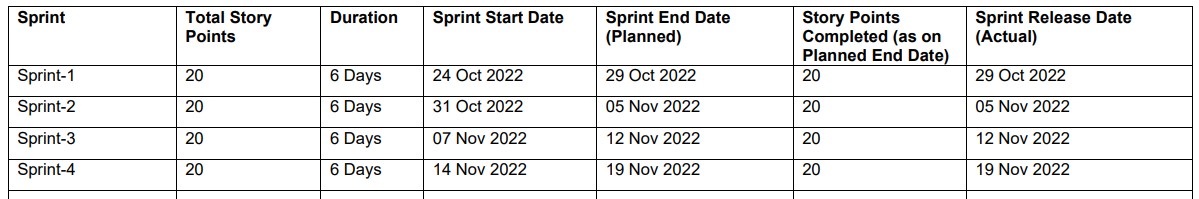


# 6. PROJECT PLANNING & SCHEDULING

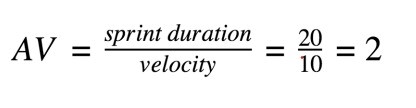
## 6.1 SPRINT PLANNING & ESTIMATION



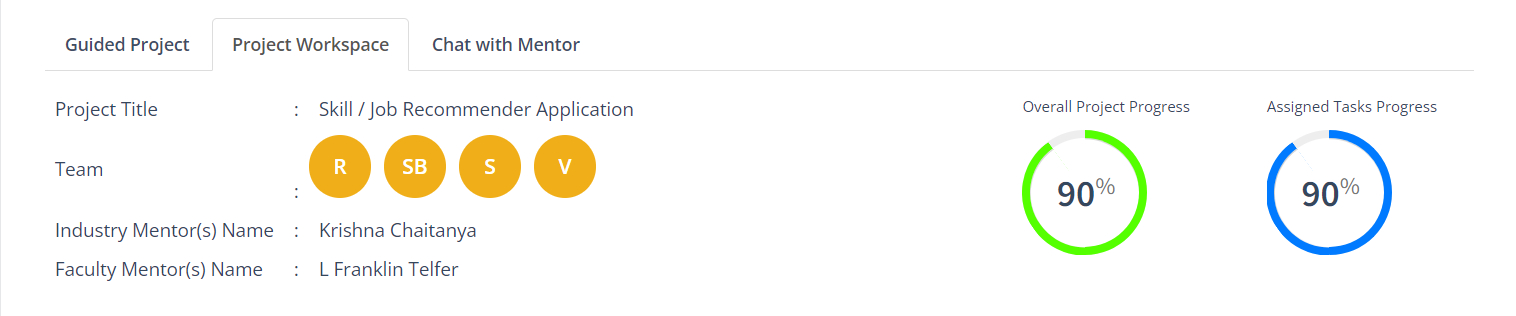
## 6.2 SPRINT DELIVERY SCHDULE



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



### 6.3 REPORTS FROM JIRAYA

****

# 7. CODING & SOLUTIONING

## 7.1 FEATURE 1

### LOGIN

|  |
| --- |
| @app.route('/login',methods = ['GET']) def login(): global userid msg = '' if request.method == 'GET':  email = request.args.get("email") password = request.args.get("password")  sql = "SELECT \* FROM USERDETAILS WHERE email = ? AND password = ?" stmt = ibm\_db.prepare(db.conn,sql)  ibm\_db.bind\_param(stmt,1,email) ibm\_db.bind\_param(stmt,2,password) ibm\_db.execute(stmt) account = ibm\_db.fetch\_assoc(stmt) print(account) if account:  return redirect(url\_for("index.html")) else:  msg = 'Invalid details. Please check the Email ID - Password combination.!' return render\_template("home.html",msg=msg) |

## SIGNUP

|  |
| --- |
| #sign-up  @app.route('/register', methods=['GET','POST']) def register(): if request.method == 'POST': name = request.form['name'] phone = request.form['Phone'] email = request.form['email'] password = request.form['password'] sql = "SELECT \* FROM USERDETAILS WHERE email = ?" stmt = ibm\_db.prepare(db.conn, sql) ibm\_db.bind\_param(stmt,1,email) ibm\_db.execute(stmt) account = ibm\_db.fetch\_assoc(stmt) print(account) if account:  msg='Job Recommender Account Already exist.kindly login!' return render\_template("home.html",msg=msg)  else:  sql ="INSERT INTO USERDETAILS(NAME,PHONE,EMAIL,PASSWORD)  VALUES('{0}','{1}','{2}','{3}')"  res = ibm\_db.exec\_immediate(db.conn,sql.format(name,phone,email,password)) msg='registration successful' return render\_template("home.html",msg=msg) |

return render\_template("register.html")

## FEATURE 2

|  |
| --- |
| from sendgrid import SendGridAPIClient from sendgrid.helpers.mail import Mail  message = Mail( from\_email='navnoble910@gmail.com', to\_emails='navnoble910@gmail.com', subject='Sending with Twilio SendGrid is Fun', html\_content='<strong>and easy to do anywhere, even with Python</strong>') try:  sg = SendGridAPIClient('SG.bvY2rix4S963KLbOiu\_dVg.I2E7DAjk8fwsrxByoNsSM0N0BLKJdKH8J88BFQZt08') response = sg.send(message) print(response.status\_code) print(response.body) print(response.headers) except Exception as e:  print(e) |

# 8. TESTING

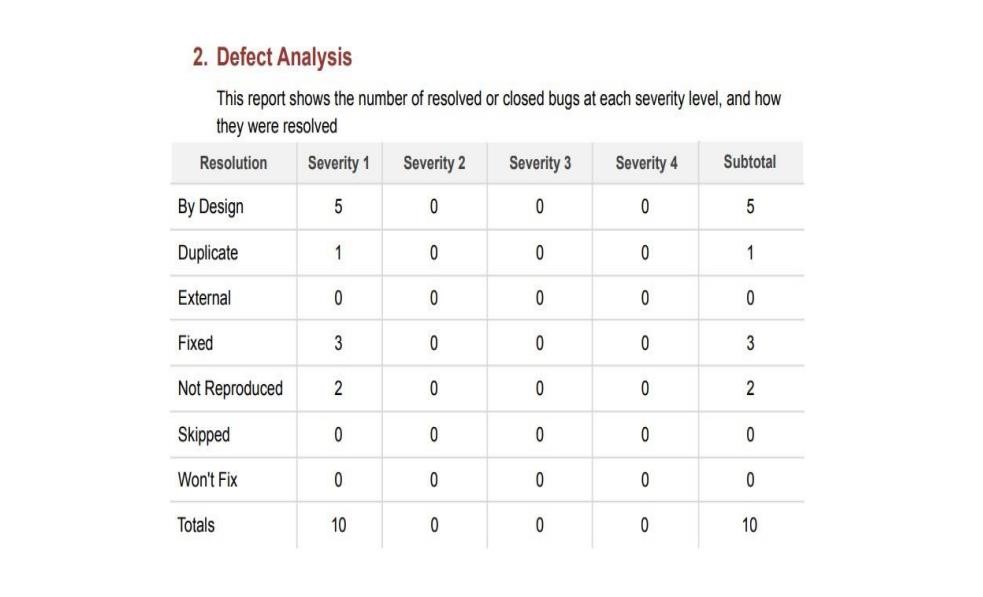
## Test Scenarios

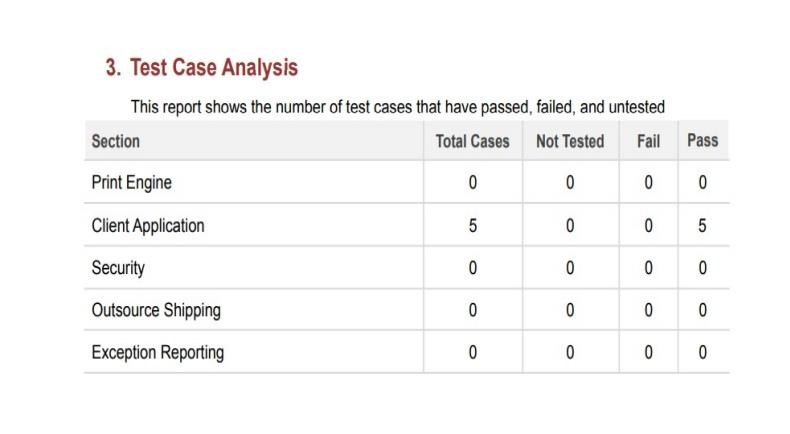
1. Verify user is able to see login page
2. Verify user is able to login to application or not?
3. Verify user is able to navigate to create your account page?
4. Verify user is able to recovery password 5 Verify login page elements

## 8.2 USER ACCEPTANCE TESTING

### 1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the [ProductName] project at the time of the release to User Acceptance Testing (UAT).





## 9. ADVANTAGES & DISADVANTAGES

### 1. ADVANTAGES

The project is identified by the merits of the system offered to the user. The merits of this project are as follows; -

* It's a web-enabled project.
* This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity
* The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
* Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records. Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer e0tent.
* User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
* From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is" we can sat that the project is user friendly which is one of the primary concerns of any good project.
* Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
* Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.
* Allocating of sample results becomes much faster because at a time the user can see the records of last years.
* Easier and faster data transfer through latest technology associated with the computer and communication.
* Through these features it will increase the efficiency, accuracy and transparency.

### 2. DISADVANTAGES

* Wrong inputs will affect the project outputs.
* Internet Connection is mandatory.
* unverified data can cause problems.

# 10. CONCLUSION

This project proved good for me as it provided practical knowledge of not only programming in ASP.NET and VB.NET web based application and no some extent windows Application and SQL Server, but also about all handling procedure related with "Skills and Job Recommender Application". It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

## 11. FUTURE SCOPE

Skills And Job Recommender Application is a web application to build such a way that. It is aimed at providing the best of opportunities to the aspirants on the grounds of their skillsets and talents. It focuses on diversifying the job application market to an extent where there will be opportunities for one and all making sure to be righteous and precise in providing services to the people who use them.

## 12. SOURCE CODE

|  |
| --- |
| import os from flask import Flask  from flask import url\_for ,render\_template,request,redirect import db  from db import ibm\_db app = Flask(\_\_name\_\_)  app.secret\_key ='AAnbb123--++'  @app.route('/') def home():  return render\_template("index.html")  @app.route('/about') def about\_us():  return render\_template("about.html")  @app.route('/category') def category():  return render\_template("category.html")  @app.route('/joblist') def joblist():  return render\_template("job-list.html") |

|  |
| --- |
| @app.route('/jobdetail') def jobdetail():  return render\_template("job-detail.html")  @app.route('/testimonial') def testimonial(): return render\_template("testimonial.html")  @app.route('/404') def fourohfour():  return render\_template("404.html") def contact():  return render\_template("contact.html")  #login  @app.route('/login',methods = ['GET']) def login(): global userid msg = '' if request.method == 'GET':  email = request.args.get("email") password = request.args.get("password")  sql = "SELECT \* FROM USERDETAILS WHERE email = ? AND password = ?" stmt = ibm\_db.prepare(db.conn,sql)  ibm\_db.bind\_param(stmt,1,email) ibm\_db.bind\_param(stmt,2,password) ibm\_db.execute(stmt) account = ibm\_db.fetch\_assoc(stmt) print(account) if account:  return redirect(url\_for("index.html")) else:  msg = 'Invalid details. Please check the Email ID - Password combination.!' return render\_template("home.html",msg=msg)  #sign-up  @app.route('/register', methods=['GET','POST']) def register():  if request.method == 'POST': name = request.form['name'] phone = request.form['Phone'] email = request.form['email'] password = request.form['password'] sql = "SELECT \* FROM USERDETAILS WHERE email = ?" stmt = ibm\_db.prepare(db.conn, sql) ibm\_db.bind\_param(stmt,1,email) ibm\_db.execute(stmt) account = ibm\_db.fetch\_assoc(stmt) print(account) if account:  msg='Job Recommender Account Already exist.kindly login!' return render\_template("home.html",msg=msg)  else: |
| sql ="INSERT INTO USERDETAILS(NAME,PHONE,EMAIL,PASSWORD)  VALUES('{0}','{1}','{2}','{3}')"  res = ibm\_db.exec\_immediate(db.conn,sql.format(name,phone,email,password)) msg='registration successful' return render\_template("home.html",msg=msg)  return render\_template("register.html")  if(\_\_name\_\_=='\_\_main\_\_'): port = os.environ.get("PORT",5000) app.run(debug=True) |

## GITHUB LINK

### https://github.com/IBM-EPBL/IBM-Project-22849-1659859395