INTRODUCTION

1.1 Project Overview

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

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

1.2 Purpose

The purpose of the projest is to create a webs

2. LITERATURE SURVEY

2.1 Existing problem

- Users can be a student or an experienced professional who needs jobs based on their skills.
- In pre-existing solutions, fake jobs are getting more and more common now-a-days.It is important because a fake job offer often leads to loss of time and money for the user.
- When the user searches for a job they may end up getting a low paid job for what they are capable of doing.
- Users may find it hard to search through a list of jobs. But, there is a chat-bot to assist the user in job recommendation.
- We eliminate this problem by regulating the salaries for the job postings on our platform.
- Users don't want to be overwhelmed by the features of the application.
- Users generally search for the job updates when they update their skill sets which match their preferences.
- Here the users get an alert based on their recent skill sets and their preferences.

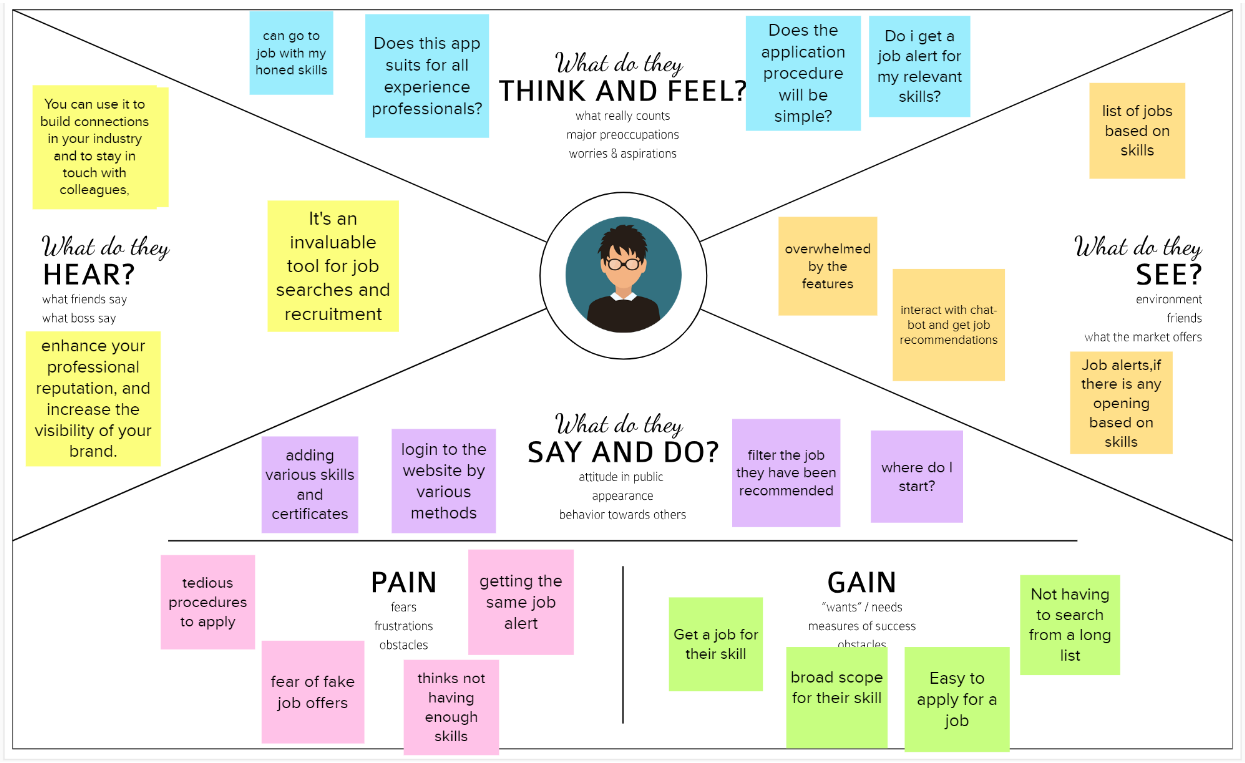
2.2 Reference

2.3 Problem Statement Definition

- User's hardship to search through jobs listed in the job-searching website.
- Unavailability of jobs or unaware of jobs that are worth applying for.
- Fear of fake job-offers and not having proper knowledge of the company.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas


What do they THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
can go to job with my honed skills
Does this app suits for all experience professionals?
Does the application procedure will be simple?
Do i get a job alert for my relevant skills?
What do they HEAR?
what friends say
what boss say
You can use it to build connections in your industry and to stay in touch with colleagues.
It's an invaluable tool for job searches and recruitment
enhance your professional reputation, and increase the visibility of your brand.
What do they SEE?
environment
friends
what the market offers
list of jobs based on skills
overwhelmed by the features
interact with chat-bot and get job recommendations
Job alerts,if there is any opening based on skills
What do they SAY AND DO?
attitude in public
appearance
behavior towards others
adding various skills and certificates
login to the website by various methods
filter the job they have been recommended
where do I start?
PAIN
fears
frustrations
obstacles
tedious procedures to apply
getting the same job alert
fear of fake job offers
thinks not having enough skills
GAIN
"wants" / needs
measures of success
obstacles
Get a job for their skill
broad scope for their skill
Easy to apply for a job
Not having to search from a long list

3.2 Ideation & Brainstorming

- Users get an alert when they have relevant skills for a job offer.
- Users can interact with chat-bot to get to know the best job-offers for their honed skills.
- Fake job-offers are eliminated via api call

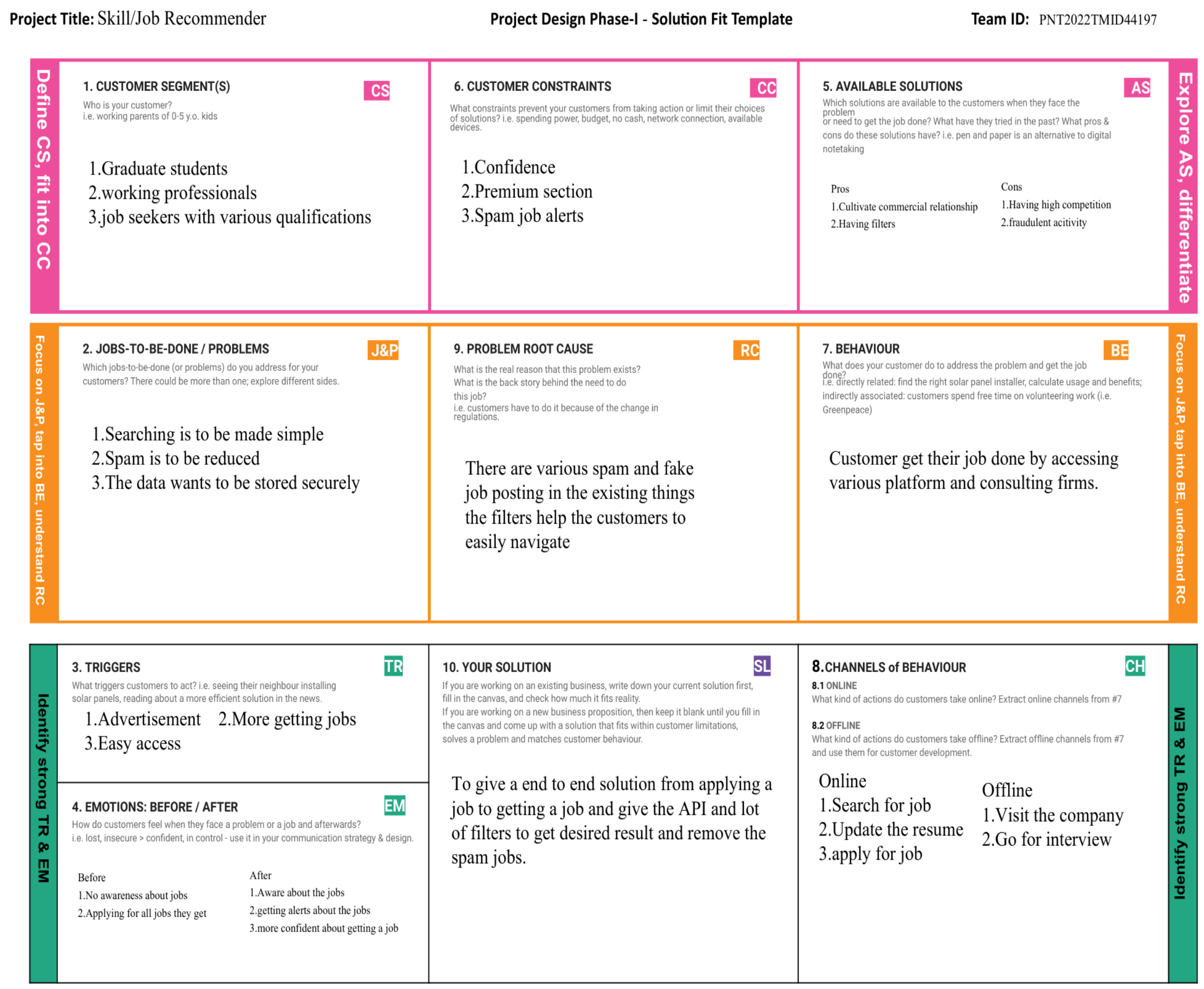
3.3 Proposed Solution

- Application uses IBM Watson chat-bot as an interactive service.
- Having a help-line 24/7 is not needed while we have AI powered chat-bots.
- With the help of chat-bots we provide an interactive service to users.
- At the end of the day, customer's feedback about the application is necessary.
- Application uses Docker to containerize it and IBM cloud for

deployment and maintenance.As we are dealing with jobs, Implementing this will increase the trust among the people.

- Feedback provides an opportunity to build a 2-way communication channel with your users.
- With the amount of users increase, during the growth of the application.
- We can provide premium features to the user with advanced option.

3.4 Problem Solution fit



Project Title: Skill/Job Recommender

Project Design Phase-I - Solution Fit Template

Team ID: PNT2022TMID44197

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

1.Graduate students
2.working professionals
3.job seekers with various qualifications

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

1.Confidence
2.Premium section
3.Spam job alerts

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem
or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

Pros	Cons
1.Cultivate commercial relationship	1.Having high competition
2.Having filters	2.fraudulent acitivity

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

1.Searching is to be made simple
2.Spam is to be reduced
3.The data wants to be stored securely

9. PROBLEM ROOT CAUSE RC

What is the real reason that this problem exists?
What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

There are various spam and fake job posting in the existing things the filters help the customers to easily navigate

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

Customer get their job done by accessing various platform and consulting firms.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

1.Advertisement 2.More getting jobs
3.Easy access

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before	After
1.No awareness about jobs	1.Aware about the jobs
2.Applying for all jobs they get	2.getting alerts about the jobs
	3.more confident about getting a job

10. YOUR SOLUTION SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

To give a end to end solution from applying a job to getting a job and give the API and lot of filters to get desired result and remove the spam jobs.

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE
What kind of actions do customers take online? Extract online channels from #7

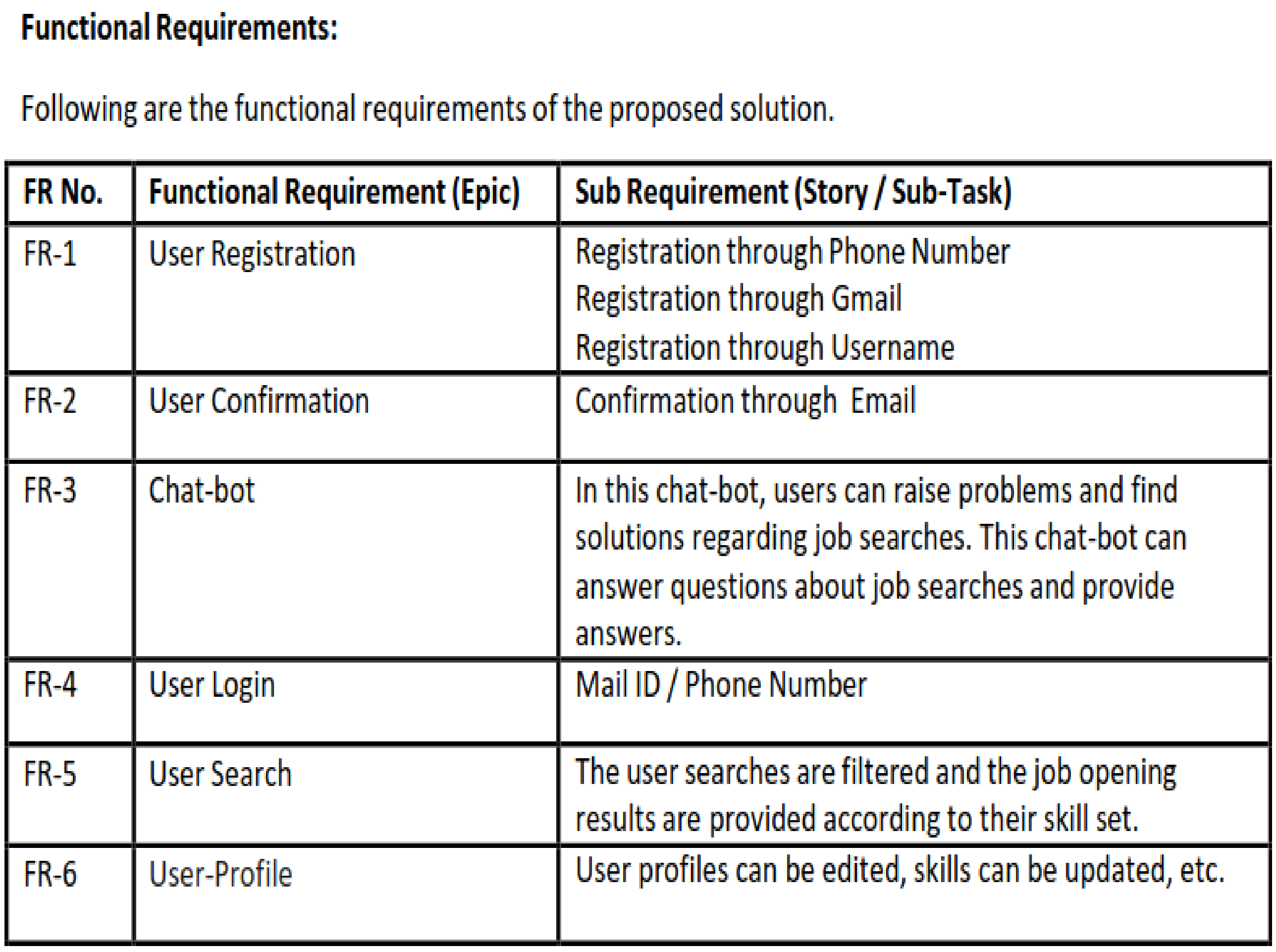
8.2 OFFLINE
What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

Online	Offline
1.Search for job	1.Visit the company
2.Update the resume	2.Go for interview
3.apply for job	

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

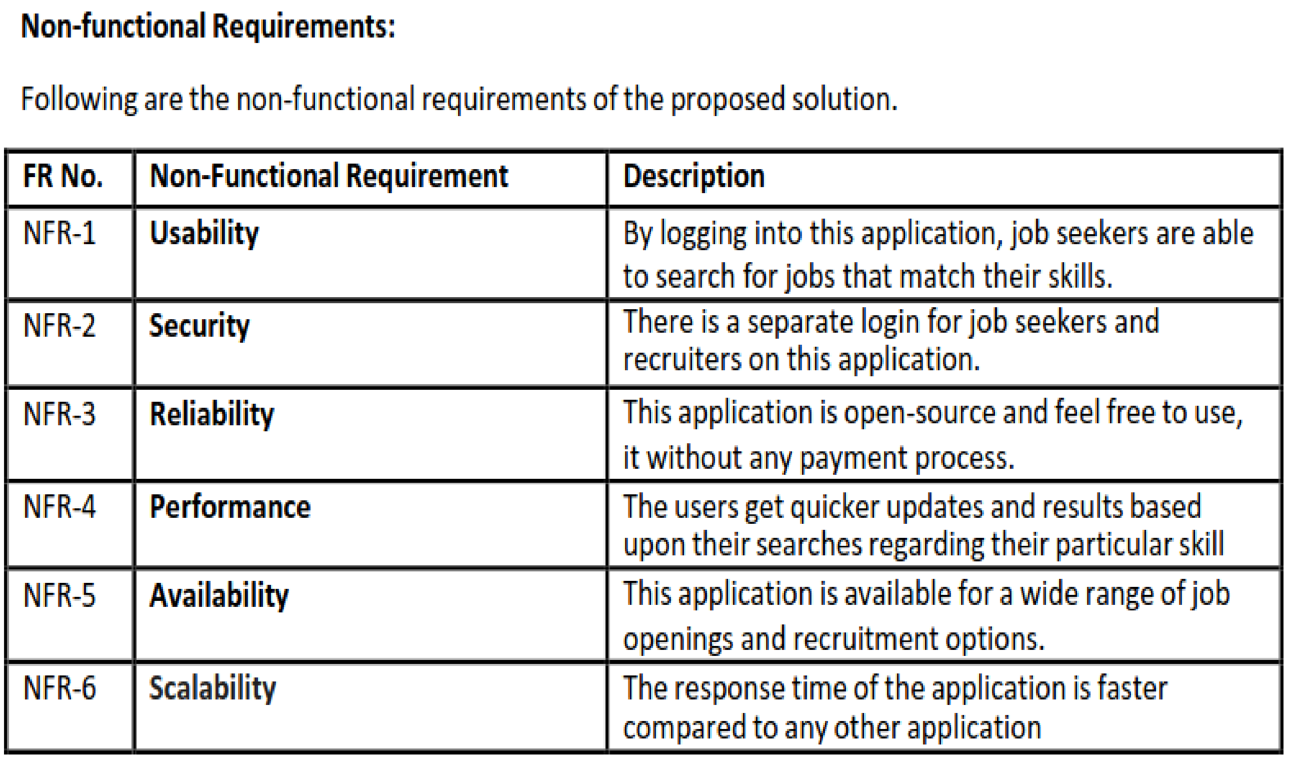
4.1 Functional requirement



Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Phone Number Registration through Gmail Registration through Username
FR-2	User Confirmation	Confirmation through Email
FR-3	Chat-bot	In this chat-bot, users can raise problems and find solutions regarding job searches. This chat-bot can answer questions about job searches and provide answers.
FR-4	User Login	Mail ID / Phone Number
FR-5	User Search	The user searches are filtered and the job opening results are provided according to their skill set.
FR-6	User-Profile	User profiles can be edited, skills can be updated, etc.



Non-functional Requirements:

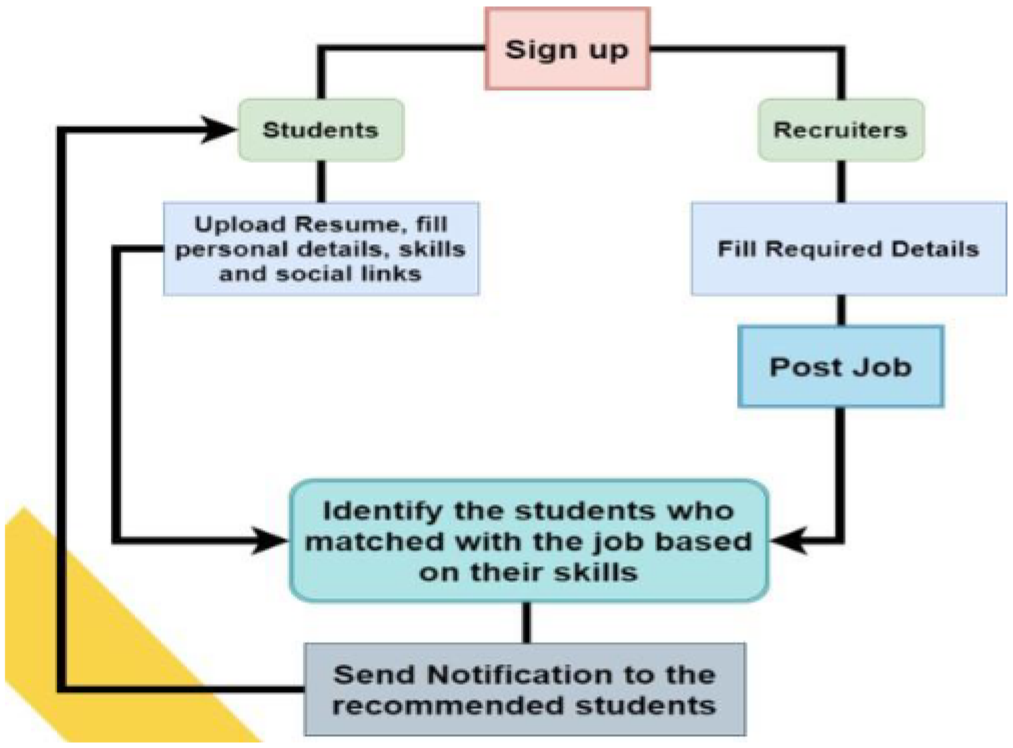
Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	By logging into this application, job seekers are able to search for jobs that match their skills.
NFR-2	**Security**	There is a separate login for job seekers and recruiters on this application.
NFR-3	**Reliability**	This application is open-source and feel free to use, it without any payment process.
NFR-4	**Performance**	The users get quicker updates and results based upon their searches regarding their particular skill
NFR-5	**Availability**	This application is available for a wide range of job openings and recruitment options.
NFR-6	**Scalability**	The response time of the application is faster compared to any other application

5. PROJECT DESIGN

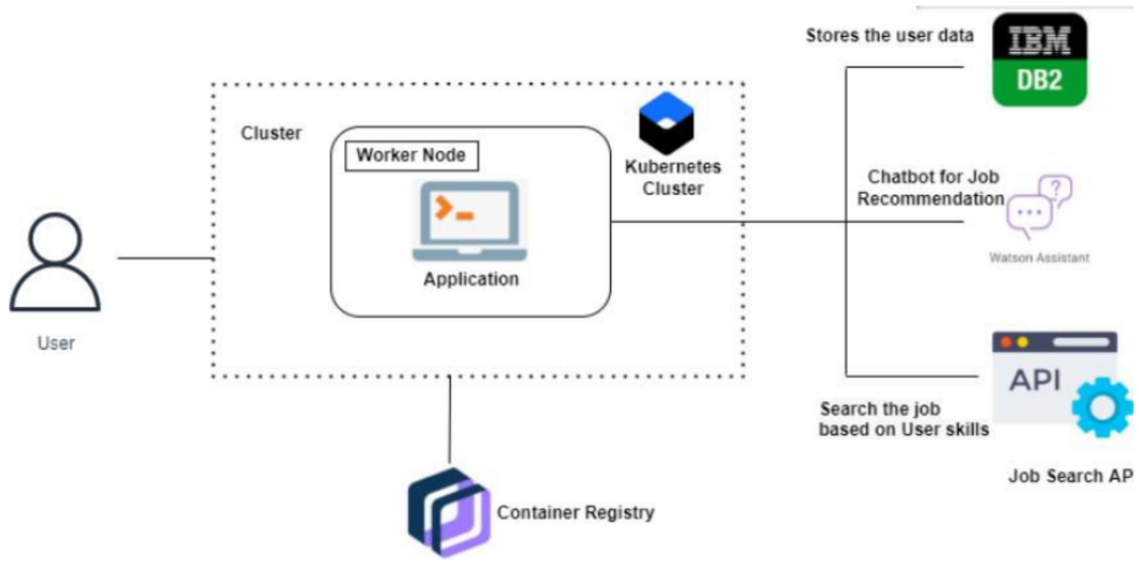
5.1 Data Flow Diagrams

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.


Sign up
Students
Recruiters
Upload Resume, fill personal details, skills and social links
Fill Required Details
Post Job
Identify the students who matched with the job based on their skills
Send Notification to the recommended students

5.2 Solution & Technical Architecture

Technical Architecture:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, and password, and confirming my password.	I can access my account/dashboard	High	Sprint-1
		USN-2	As a user, I will receive a confirmation email once I have registered for the application.	I can receive a confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail and Google Account.	I can register & access the dashboard with Gmail Login	Medium	Sprint-2
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-2
	Dashboard	USN-5	As a user, I can access my dashboard after signing in.	I can access my account/ dashboard	High	Sprint-3

Customer (Web user)	Access	USN-6	As a user, I can set up a profile, and basic details by signing in.			
		USN-7	As a user, I will upload my resume, certificates, and other technical requirements.	I can perform several tasks in the application	Medium	Sprint-3
Customer Care Executive	Chat bot	USN-8	As a user, I can seek guidance from the customer care executive.		High	Sprint-4
Administrator	DBMS	USN-9	As an administrator, I can keep the applications of your organization relies on running	I can perform various modifications in the applications	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

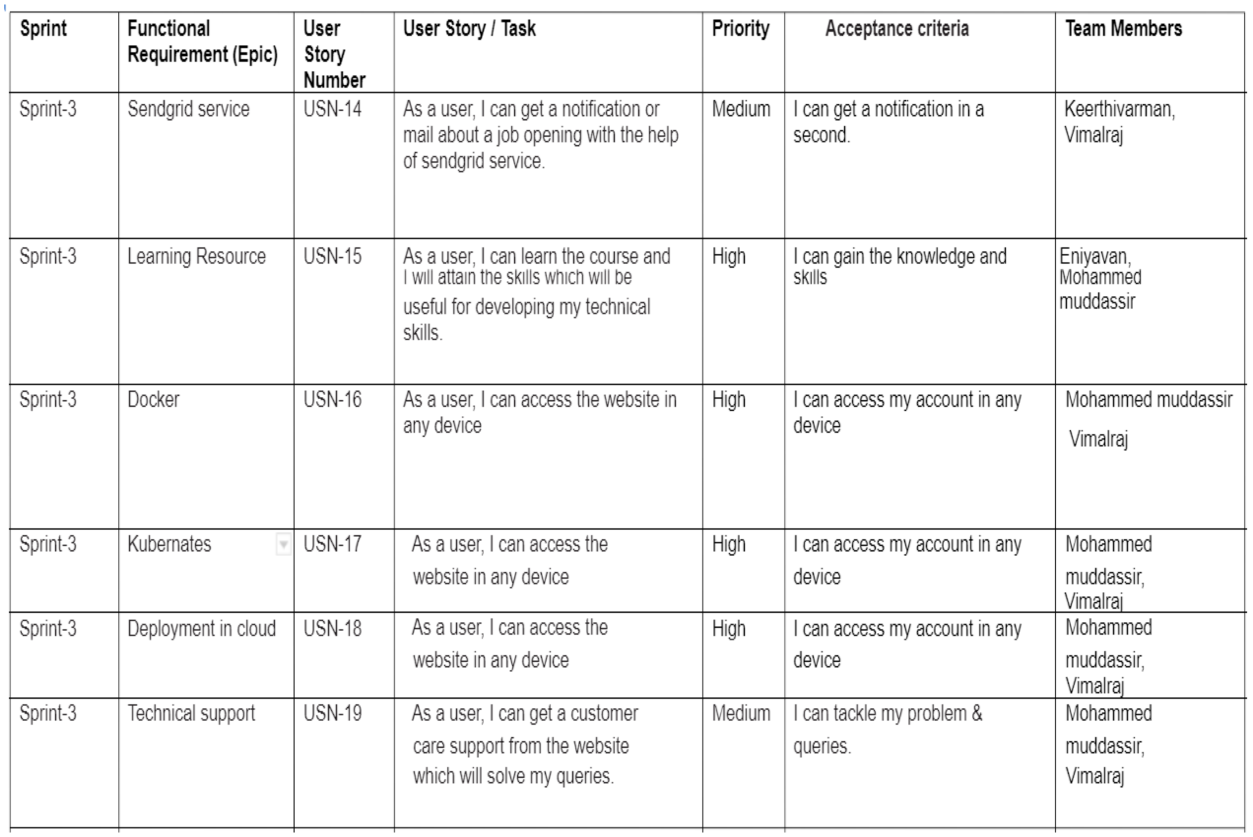
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Priority**	**Acceptance criteria**	**Team Members**
Sprint-1	UI Design	USN-1	As a user, I can see and experience an awesome user interface in	Medium	Better Impression about a website	Keerthivarman, Vimalraj

			the website			
Sprint-1	Registration	USN-2	As a user, I can register for the application by entering my email, password, and confirming my password.	High	I can access my account / dashboard	Keerthivarman, Mohammed muddassir
Sprint-1		USN-3	As a user, I will receive confirmation email once I have registered for the application	High	I can receive confirmation email & click confirm	Eniyavan, Vimalraj
Sprint-1		USN-4	As a user, I can register for the application through Facebook	Low	I can register & access the dashboard with Facebook Login	Mohammed muddassir, Vimalraj
Sprint-1		USN-5	As a user, I can register for the application through Gmail	Medium	I can receive confirmation email & click confirm	Keerthivarman, Mohammed muddassir
Sprint-1	Login	USN-6	As a user, I can log into the application	High	I can access my account /	Keerthivarman,

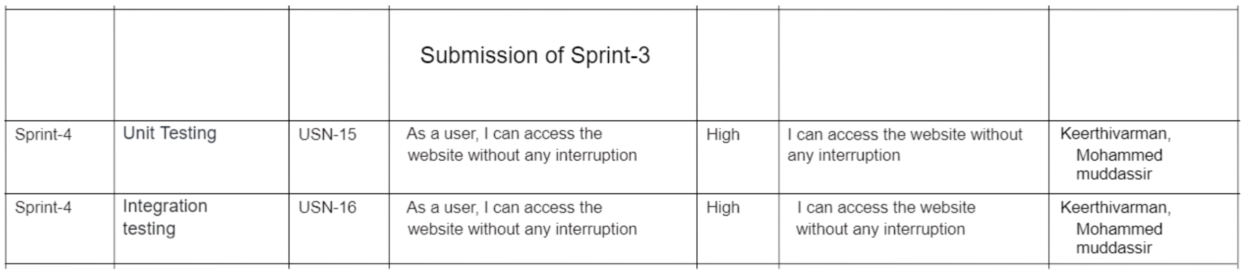
			by entering email & password		dashboard	Vimalraj
Sprint-!	Flask	USN-7	As a user, I can access the website in a second	High	I can access my account / dashboard	Mohammed muddassir, Vimalraj

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-1	Dashboard	USN-8	As a user, If I Logged in correctly, I can view my dashboard and I can navigate to any pages which are already listed there.	High	I can access all the pages/ dashboard	Keerthivarman, Eniyavan
			Submission Of Sprint-1			
Sprint-2	User Profile	USN-9	As a user, I can view and update my details	Medium	I can modify my details/data	Keerthivarman, Vimalraj

Sprint-2	Cloud Storage	USN-11	As a user, I can upload my photo, resume and much more in the website.	Medium	I can Upload my documents and details	Keerthivarman, Eniyavan
Sprint-2	Chatbot	USN-12	As a user, I can ask the Chatbot about latest job openings, which will help me and show the recent job openings based on my profile	High	I can know the recent job openings	Keerthivarman, Mohammed muddassir
Sprint-2	Identity-Aware	USN-13	As a User, I can access my account by entering by correct login credentials. My user credentials is only displayed to me.	High	I can have my account safely	Keerthivarman, Mohammed muddassir



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

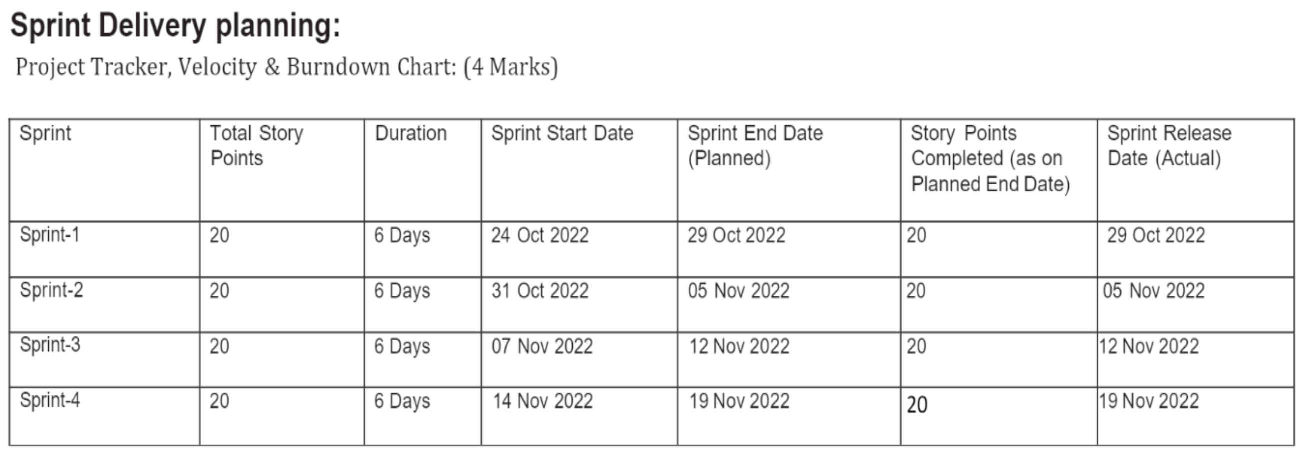


			Submission of Sprint-3			
Sprint-4	Unit Testing	USN-15	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Keerthivarman, Mohammed muddassir
Sprint-4	Integration testing	USN-16	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Keerthivarman, Mohammed muddassir

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Priority	Acceptance criteria	Team Members
Sprint-4	System testing	USN-17	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Keerthivarman, Mohammed muddassir
Sprint-4	Correction	USN-18	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Keerthivarman, Mohammed

						muddassir
Sprint-4	Acceptance testing	USN-19	As a user, I can access the website without any interruption	High	I can access the website without any interruption	Keerthivarman, Mohammed muddassir
			Submission of Sprint-4			

6.2 Sprint Delivery Schedule



Sprint Delivery planning:

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

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

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

app.py

```
  from flask import Flask,render_template,request,session,
redirect,url_for
  import ibm_db
  app = Flask(__name__)
  app.secret_key='vy@ur434'

  conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=19af64
46-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.clo
ud;PORT=30699;SECURITY=SSL;SSLServerCertificate=D
igiCertGlobalRootCA.crt;UID=bts04620;PWD=3m4fzYEma
QCgXqGx",'','')
  print("connection successful...")
  @app.route('/')
  def home():
     return render_template('index.html')

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

  @app.route('/forgot')
  def forgot():
     return render_template('forgotten-password.html')

  @app.route('/signup', methods=['POST','GET'])
  def signup():
      if request.method == 'POST':
           try:
              sql = "INSERT INTO users 
VALUES('{}','{}','{}','{}')".format(request.form["name"],re
quest.form["email"],request.form["phone"],request.form["
password"])
              ibm_db.exec_immediate(conn,sql)
              #flash("successfully Registered !")
              return render_template('login.html')
           except:
              #flash("Account already exists! ")
              return render_template('signup.html')
      else:
              return render_template('signup.html')

  @app.route('/login', methods=['POST','GET'])
  def login():
       if request.method == 'POST':
          email = request.form["email"]
          password = request.form["password"]
          sql = "SELECT COUNT(*) FROM users WHERE 
EMAIL=? AND PASSWORD=?"
          stmt = ibm_db.prepare(conn,sql)
          ibm_db.bind_param(stmt, 1, email)
          ibm_db.bind_param(stmt, 2, password)
```

```
            ibm_db.execute(stmt)
            res = ibm_db.fetch_assoc(stmt)
            if res['1'] == 1:
                session['loggedin'] = True
                session['email'] = email
                return render_template('userpage.html')
            else:
                return render_template('login.html')
        else:
                return render_template('login.html')

@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('username', None)
    return redirect(url_for('login'))

if __name__=='__main__':
    app.config['SESSION_TYPE']= 'filesystem'
    app.run(debug=True)
```

HTML CODES

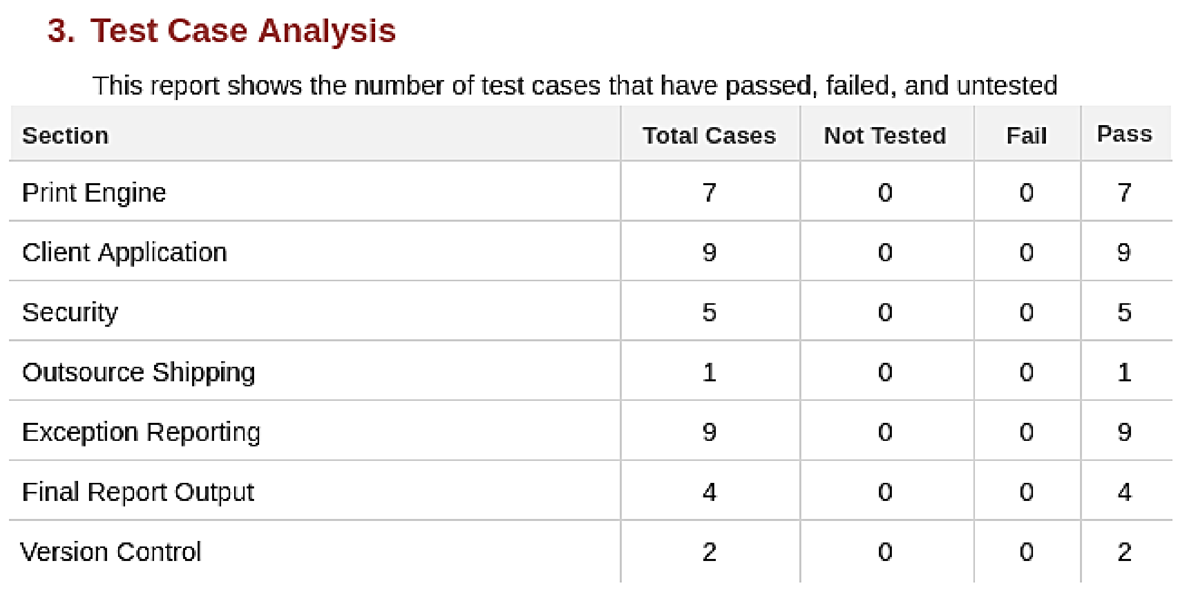
https://github.com/IBM-EPBL/IBM-Project-42835-1660710068/tree/main/Project%20Development%20Phase/Sprint%204/templates

7.1 Feature 1

- Chatbot

8. TESTING

8.1 Test Cases

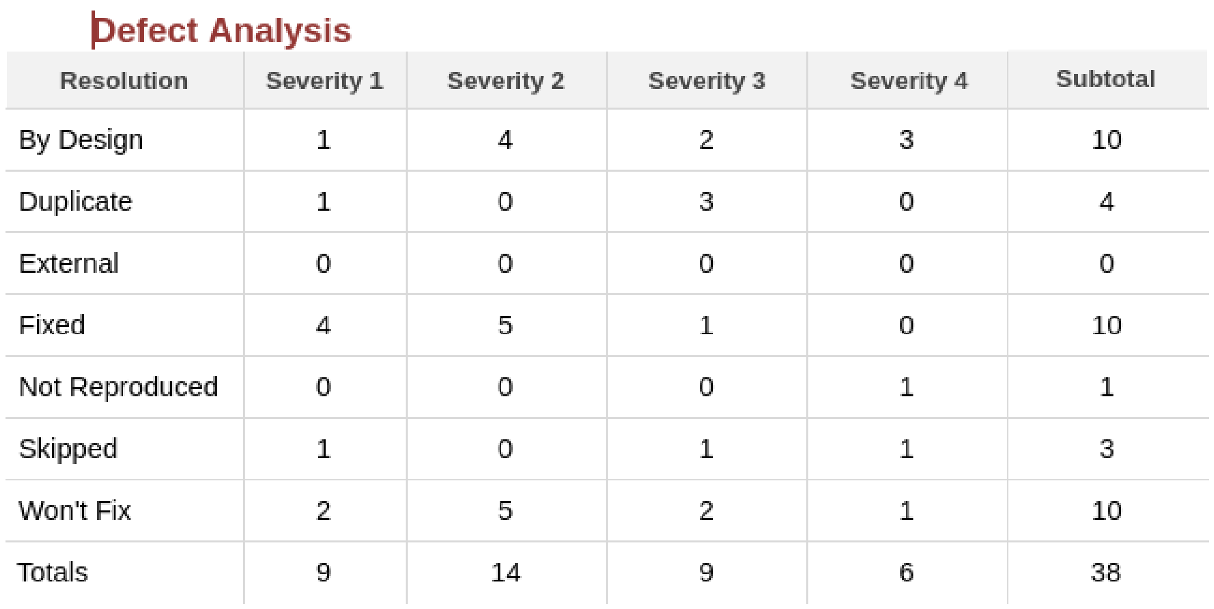


3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	9	0	0	9
Security	5	0	0	5
Outsource Shipping	1	0	0	1
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

8.2 User Acceptance Testing



Defect Analysis

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	4	2	3	10
Duplicate	1	0	3	0	4
External	0	0	0	0	0
Fixed	4	5	1	0	10
Not Reproduced	0	0	0	1	1
Skipped	1	0	1	1	3
Won't Fix	2	5	2	1	10
Totals	9	14	9	6	38

9. RESULTS

9.1 Performance Metrics

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- User will save time and money.
- User will receive qualified, quality candidates.
- Retention rate is typically better.

DISADVANTAGES

- You may get a recommendation based on bias.
- Employee referrals can invite opportunity for negative company politics.

11. CONCLUSION

We have seen from our literature review and from the challenges that faced the holistic e-recruiting platforms, an increased need for enhancing the quality of candidates/job matching. The recommender system technologies accomplished significant success in a broad range of applications and potentially a powerful searching and recommending techniques. Consequently, there is a great opportunity for applying these technologies in recruitment environment to improve the matching quality. 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. We presented state of the art of job recommendation as well as, a comparative study for its approaches that proposed by literatures. We conclude that the field of job recommendations is still unripe and require further improvements. As part of our ongoing research, we aim to build a new recommendation approach and test with real data for employee and staffing data from large companies.

12. FUTURE SCOPE

The objective of recommender systems is to provide recommendations based on recorded information on the users' preferences. These systems use information filtering techniques to process information and provide the user with potentially more relevant items.

13. APPENDIX

Source Code

app.py

```
    from flask import Flask,render_template,request,session,
redirect,url_for
 import ibm_db


 app = Flask(__name__)
 app.secret_key='vy@ur434'

 conn =
ibm_db.connect("DATABASE=bludb;HOSTNAME=19af64
46-6171-4641-8aba-
9dcff8e1b6ff.c1ogj3sd0tgtu0lqde00.databases.appdomain.clo
ud;PORT=30699;SECURITY=SSL;SSLServerCertificate=D
igiCertGlobalRootCA.crt;UID=bts04620;PWD=3m4fzYEma
QCgXqGx",'','')
 print("connection successful...")
 @app.route('/')
 def home():
    return render_template('index.html')

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

 @app.route('/forgot')
 def forgot():
    return render_template('forgotten-password.html')

 @app.route('/signup', methods=['POST','GET'])
 def signup():
    if request.method == 'POST':
        try:
          sql = "INSERT INTO users
VALUES('{}','{}','{}','{}')".format(request.form["name"],re
quest.form["email"],request.form["phone"],request.form["
password"])
          ibm_db.exec_immediate(conn,sql)
          #flash("successfully Registered !")
          return render_template('login.html')
        except:
           #flash("Account already exists! ")
           return render_template('signup.html')
   else:
           return render_template('signup.html')

 @app.route('/login', methods=['POST','GET'])
 def login():
    if request.method == 'POST':
       email = request.form["email"]
       password = request.form["password"]
       sql = "SELECT COUNT(*) FROM users WHERE
EMAIL=? AND PASSWORD=?"
       stmt = ibm_db.prepare(conn,sql)
       ibm_db.bind_param(stmt, 1, email)
       ibm_db.bind_param(stmt, 2, password)
       ibm_db.execute(stmt)
       res = ibm_db.fetch_assoc(stmt)
       if res['1'] == 1:
           session['loggedin'] = True
           session['email'] = email
           return render_template('userpage.html')
       else:
           return render_template('login.html')
    else:
           return render_template('login.html')
```

```
@app.route('/logout')
def logout():
    session.pop('loggedin', None)
    session.pop('username', None)
    return redirect(url_for('login'))

if __name__=='__main__':
    app.config['SESSION_TYPE']= 'filesystem'
  app.run(debug=True)
```

HTML CODE

GitHub Link

https://github.com/IBM-EPBL/IBM-Project-42835-1660710068

Project Demo Link

https://drive.google.com/drive/folders/1JAwzQlPkcWVB-jyk6Ed4WuMxe8NiUWXh?usp=sharing