



## **SKILL / JOB RECOMMENDER APPLICATION**

**NALAIYA THIRAN - PROJECT REPORT**

**PROJECT ID: PNT2022TMID00862**

*Submitted by*

**Bandi Chaithanya Krishna Reddy [211419104039]**

**K. ASHWIN [211419104026]**

**N. GOKUL [211419104081]**

**DHANUSH BOBBY REDDY [211419104058]**

*In partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**(AN AUTONOMOUS INSTITUTION , AFFILIATED TO ANNA UNIVERSITY)**

**NOVEMBER 2022**

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

**(AN AUTONOMOUS INSTITUTION , AFFILIATED TO ANNA UNIVERSITY)**

**BONAFIDE CERTIFICATE**

Certified that this project report

**“SKILL / JOB RECOMMENDER APPLICATION – PNT2022TMID00862”**

is the bonafide work of

**Bandi Chaithanya Krishna Reddy (211419104039)**

**K. ASHWIN (211419104026)**

**N. GOKUL (211419104081)**

**DHANUSH BOBBY REDDY (211419104058)**

who carried out the NALAIYA THIRAN project work under the supervision.

**Krishna Chaitanya**

**INDUSTRY MENTOR  
IBM**

**Dr.P.J. Sathish Kumar**

**FACULTY MENTOR  
Department of CSE  
Panimalar Engineering College**

# **TABLE OF CONTENTS**

<b><u>CHAPTER NO</u></b>	<b><u>TITLE</u></b>	<b><u>PG NO</u></b>
<b>1</b>	<b>INTRODUCTION</b>	
	1.1 Project Overview	<b>5</b>
	1.2 Purpose	<b>5</b>
<b>2</b>	<b>LITERATURE SURVEY</b>	
	2.1 Existing Problem	<b>5</b>
	2.2 References	<b>5</b>
	2.3 Problem Statement	<b>6</b>
<b>3</b>	<b>IDEATION &amp; PROPOSED SOLUTION</b>	
	3.1 Empathy Map Canvas	<b>6</b>
	3.2 Ideation & Brainstorming	<b>7</b>
	3.3 Proposed Solution	<b>8</b>
	3.4 Problem Solution fit	<b>9</b>
<b>4</b>	<b>REQUIREMENT ANALYSIS</b>	
	4.1 Functional Requirements	<b>10</b>
	4.2 Non-Functional Requirements	<b>11</b>
<b>5</b>	<b>PROJECT DESIGN</b>	
	5.1 Data Flow Diagrams	<b>12</b>
	5.2 Solution & Technical Architecture	<b>13</b>
	5.3 User Stories	<b>15</b>
<b>6</b>	<b>PROJECT PLANNING &amp; SCHEDULING</b>	
	6.1 Sprint Planning & Estimation	<b>15</b>
	6.2 Sprint Delivery Schedule	<b>16</b>
<b>7</b>	<b>CODING &amp; SOLUTION</b>	
	7.1 Feature 1	<b>17</b>
	7.2 Feature 2	<b>20</b>
<b>8</b>	<b>TESTING</b>	<b>23</b>

9	<b>RESULT</b>	<b>24</b>
	9.1 Performance Metrics	
	9.2 Output	
10	<b>ADVANTAGES &amp; DISADVANTAGES</b>	<b>28</b>
11	<b>CONCLUSION</b>	<b>30</b>
12	<b>FUTURE SCOPE</b>	<b>30</b>
13	<b>APPENDIX</b>	<b>30</b>

GitHub & Project Demo Link

## **Project Report**

### **1. INTRODUCTION**

#### **1.1 Project Overview**

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 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 main aim of the project is to building a model 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.

### **2. LITERATURE SURVEY**

#### **2.1 Existing problem**

Skill -based recommendation services platforms wants to turn into a leading source of the job recommender platform. The job and skill recommender system is expected to reduce unemployment and improve the skills of job seekers to boost the country's economy. Job recommender is giving guidance for job seekers to get quick contact for those jobs. Immediate announcement of the latest job openings, and walk-in interviews with different locations and offers you full details of recruitment such as the number of job vacancies, eligibility criteria, interview date, required qualifications etc.

#### **2.2 References**

<https://www.naukri.com/>

<https://in.indeed.com/>

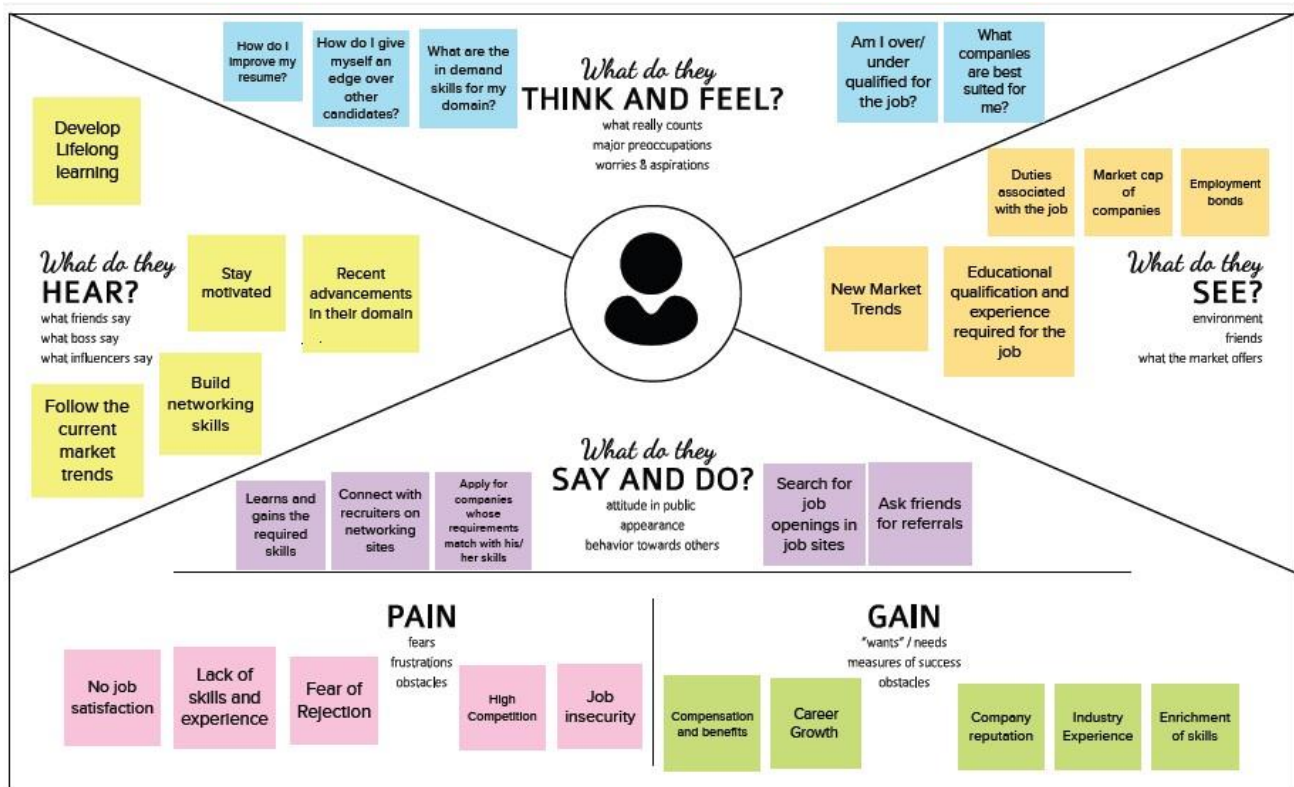
## 2.3 Problem Statement Definition

The main aim of the project is to develop an end-to-end web application capable of displaying the current job openings based on the user skill set. Users will interact with the chat-bot and can get recommendations based on their skills.

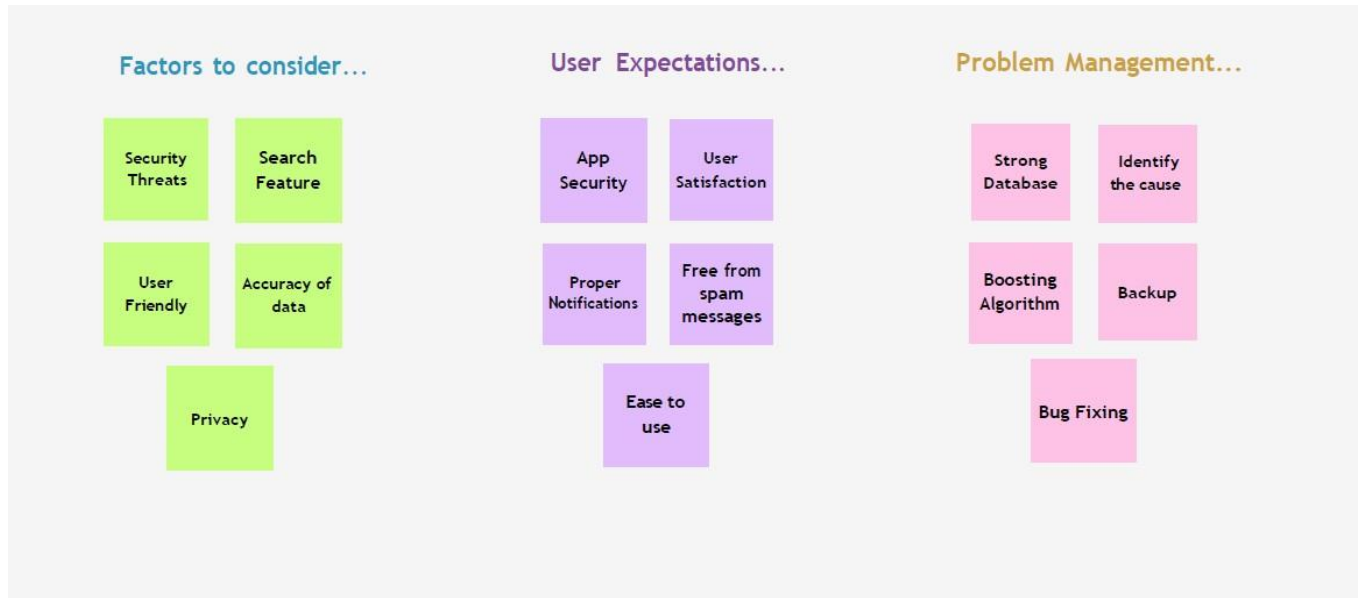
This Application uniquely identifies the user's skills recommend the job according to the user's interest.

## 3. IDEATION & PROPOSED SOLUTION

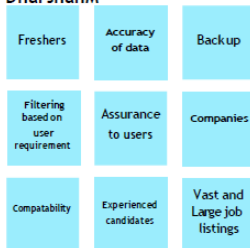
### 3.1 Empathy Map Canvas



### 3.2 Ideation & Brainstorming



**Dhirandar  
DharshanM**



**Dasvanth S**



**Hariharn  
V**



**Dineshraj  
V M**



### 3.3 Proposed Solution

<u>S.No.</u>	<u>Parameter</u>	<u>Description</u>
1.	Problem Statement (Problem to be solved)	Every industry has a lot of career opportunities, but job seekers are unaware of them. The unemployability crisis can be solved if every job seeker receives the right career guidance and proper job role training. So, to eradicate the unemployment crisis, for the job seekers to find a job they desire, match their qualifications and skills.
2.	Idea / Solution description	To develop an end-to-end web application capable of displaying the current job openings based on the user skill set. Users will interact with the chat-bot and can get recommendations based on their skills.
3.	Novelty / Uniqueness	This Application uniquely identifies the user's skills recommend the job according to the user's interest.
4.	Social Impact / Customer Satisfaction	The job and skill recommender system is expected to reduce unemployment and improve the skills of job seekers to boost the country's economy. The customer satisfaction can be measured by customer loyalty and customer reviews after deployment of the project.
5.	Business Model (Revenue Model)	A subscription model can be provided for both employees and employers with additional costs for features along with recurring monthly or yearly costs.
6.	Scalability of the Solution	It is lifelong recommender app. Once the user has login to this application, he will be notified about the job up to date.

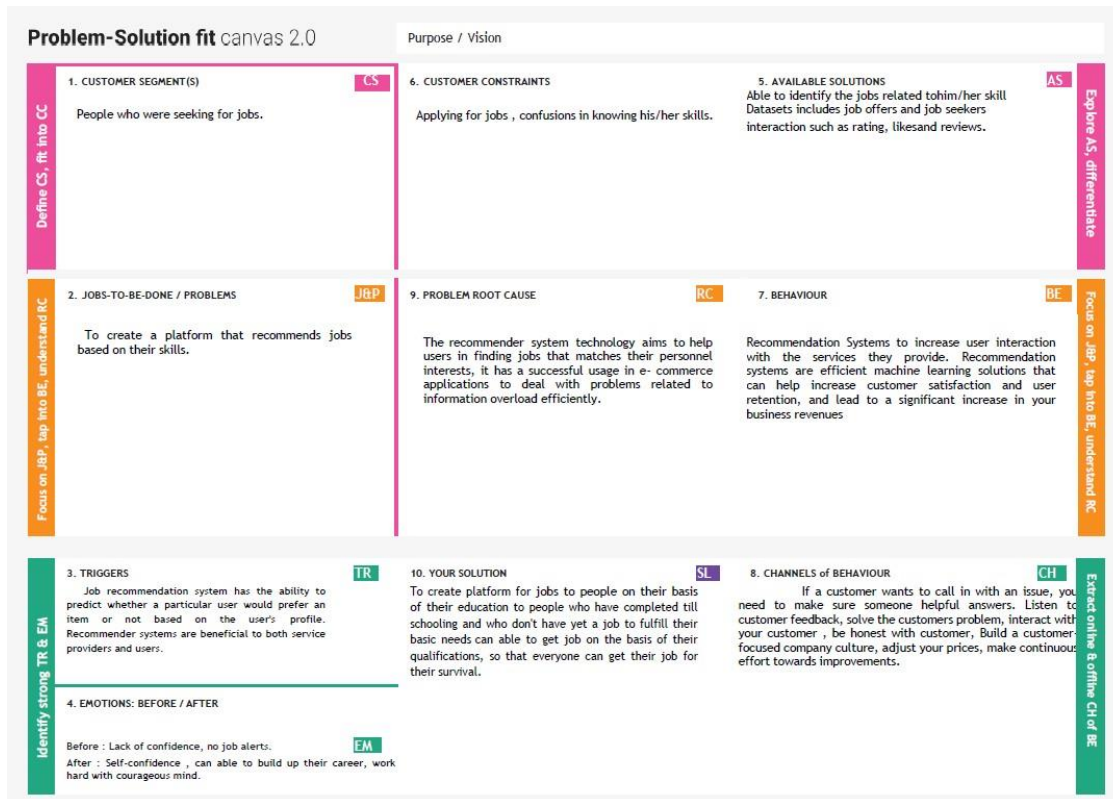


### **3.4 Problem Solution fit**

The Problem-Solution Fit simply means that you have found a problem with your customer and that the solution you have realized for it actually solves the customer's problem. It helps entrepreneurs, marketers and corporate innovators identify behavioral patterns

#### **Purpose:**

- Solve complex problems in a way that fits the state of your customers.
- Succeed faster and increase your solution adoption by tapping into existing mediums and channels of behavior.
- Sharpen your communication and marketing strategy with the right triggers and messaging.
- Increase touch-points with your company by finding the right problem-behavior fit and building trust by solving frequent annoyances, or urgent or costly problems.



## 4 REQUIREMENT ANALYSIS

### 4.1 Functional Requirement

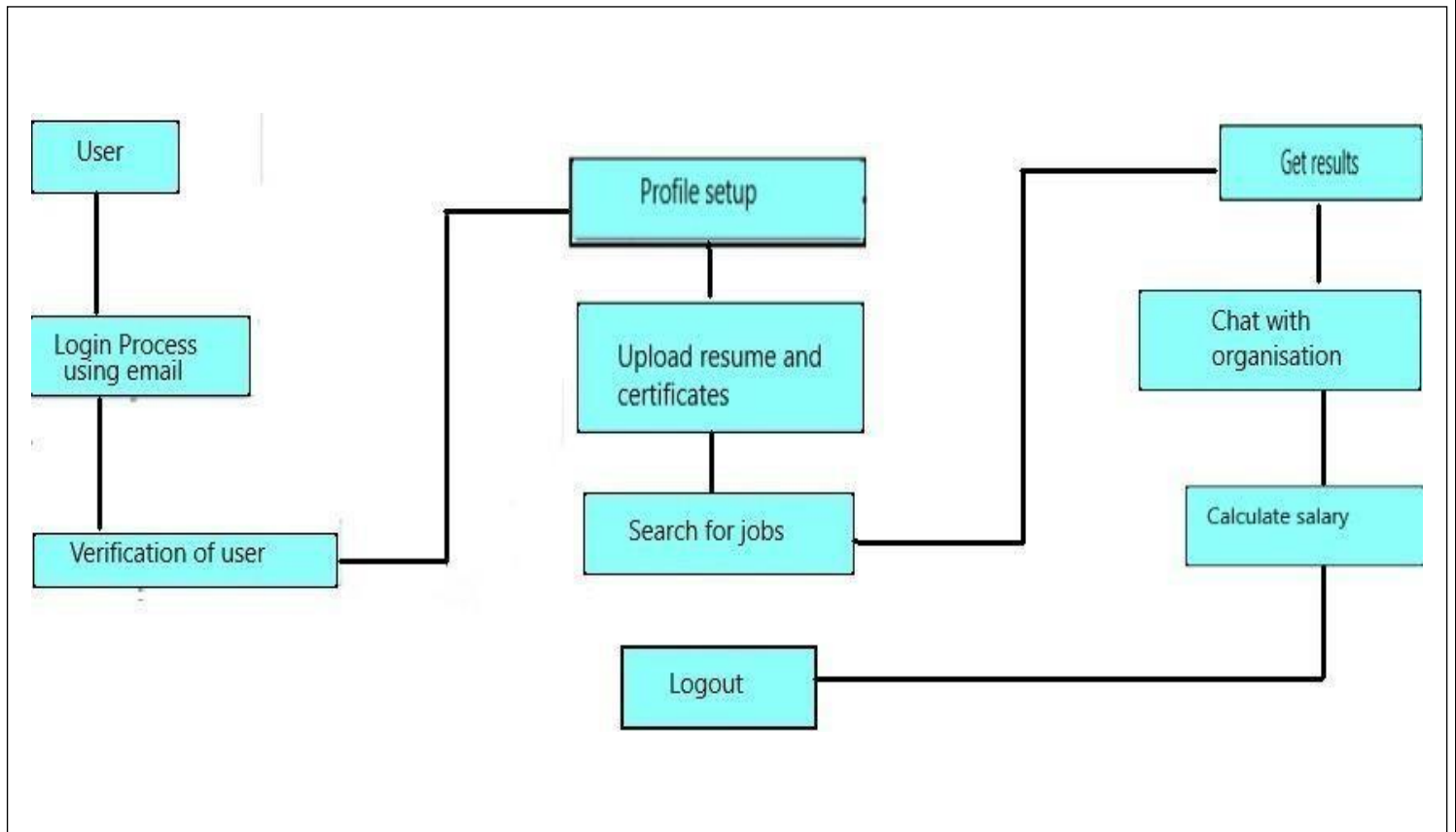
S. No	FUNCTIONAL REQUIREMENT (Epic)	SUB REQUIREMENT (Story)
1)	Sign In / Login	Register with username, password
2)	Profile Registration	Register with username, password, email, qualification, skills. This data will be stored in a database.
3)	Job profile display	Display job profiles based on availability, location ,skills
4)	Chatbot	A chat on the webpage to solve user queries and issues
5)	Job registration	A copy of the company the user applied for with its registration/description details will be sent to the registered email id.
6)	Logout	Logging out of the webpage.

### 3.4 Non-Functional requirements

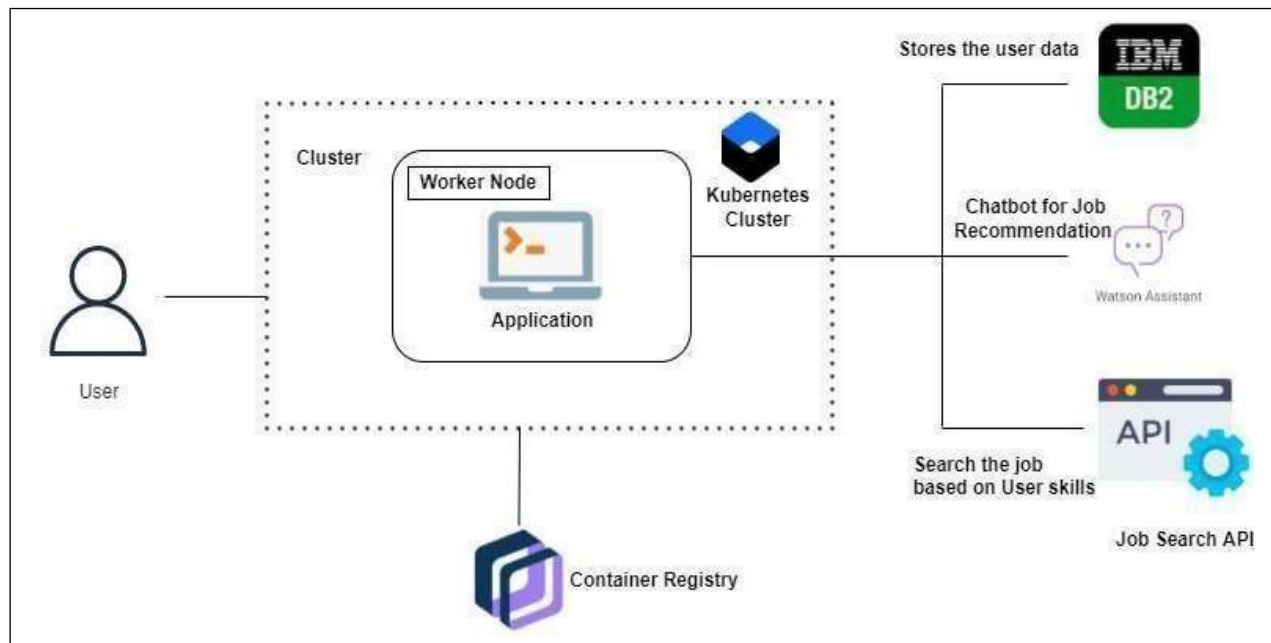
<u>S. No</u>	<u>NON-FUNCTIONAL REQUIREMENT</u>	<u>DESCRIPTION</u>
1)	Usability	The webpage will be designed in such away that any non-technical user can easily navigate through it and complete the job registration work.
2)	Security	Using of SSL certificate (Python Flask to Cloud connect) will provide security to the project. Database will be safely stored in DB2.
3)	Reliability	To make sure the webpage doesn't go down due to network traffic.
4)	Performance	Focus on loading the webpage as quickly as possible irrespective of the number of user/integrator traffic.
5)	Availability	This webpage will be available to all users at any given point of time.
6)	Scalability	Increasing the storage space of database can increase the number of users.

## 5 PROJECT DESIGN

### 5.1 Data Flow Diagram



## 5.2 Solution & Technical Architecture



<u>S.No.</u>	<u>Parameter</u>	<u>Description</u>
1.	Is the System Robust?	Yes, it is a partially buildable platform as the budget required will be more as the cloud is a pay-per-use model and the time taken will be quiet.
2.	Is it highly modifiable?	Indeed, the framework is modifiable and it can own up to the progressions by recognizing blunders that require to be fixed and new functionalities. It is exceptionally receptive to progressions.

3.	Is it Scalable?	Indeed, the framework proposed is exceptionally versatile as it can deal with the developing responsibility where great execution is likewise expected to effectively work. Organization of the stage has been finished utilizing different OS virtualization stage it will deal with the responsibility genuinely.
----	-----------------	---

### 5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail.		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password.		High	Sprint-1
	Dashboard	USN-5	As a user, I can access my dashboard after signing in.	I can access my account / dashboard	High	Sprint-1
Customer (Web user)	Access	USN-6	As a user, I can setup a profile, and basic details by signing in.			
		USN-7	As a user, I will upload my resume, certificates, and other requirements.	I can perform several task in the application	Medium	Sprint-1
Customer Care Executive	Chatbot	USN-8	As a user, I can seek guidance from the customer care executive.		High	Sprint-1
Administrator	DBMS	USN-9	As a administrator, I can keep the applications of your organization relies on running.	I can perform various modifications in the applications.	High	Sprint-1

## 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 job seeker experience an awesome user interface in the website	Medium	Better Impression about a website	Dinesh Raj
Sprint-1	Registration	USN-2	As a job seeker register for the application by entering my email, password, and confirming my password.	High	I can access my account / dashboard	Dasvanth
Sprint-1		USN-3	As a job seeker receive confirmation email once I have registered for the application	High	I can receive confirmation email & click confirm	Hari Haran
Sprint-1		USN-4	As a job seeker register for the application through Facebook	Low	I can register & access the dashboard with Facebook Login	Dharshan
Sprint-1		USN-5	As a job seeker register for the application through Gmail	Medium	I can receive confirmation email & click confirm	Dinesh Raj
Sprint-1	Login	USN-6	As a job seeker log into the application by entering email & password	High	I can access my account / dashboard	Dasvanth, Dharshan

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	Hari Haran
Sprint-2	User Profile	USN-9	As a user, I can view and update my details	Medium	I can modify my details/data	Dinesh Raj
Sprint-2	Database	USN-10	As a user, I can store my details and data in the website w	Medium	I can store my data	Dasvanth
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	Dharshan
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	Hari Haran
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	Dinesh Raj

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.	Dasvanth
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	Dharshan
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	Dinesh Raj
Sprint-3	Kubernetes	USN-17	As a user, I can access the website in any device	High	I can access my account in any device	Dasvanth
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	Hari Haran
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.	Dharshan
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	Dharshan
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	Hari Haran

## 6.2 Sprint Delivery Schedule

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)

### 7.1 Feature 1

```
1 import ibm_db
2 from flask import Flask, flash, redirect, render_template, request, url_for
3 from flask_cors import CORS, cross_origin
4 from flask import session
5 import sqlite3
6 import os
7
8 conn=ibm_db.connect("DATABASE=b1udb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-10cf081900bf.c1ogj3sd0tgtu0lqde00.databases.appdomain.cloud;PORT=32304;
9
10 con=sqlite3.connect("myimage.db")
11 con.execute("create table if not exists image(pid integer primary key,img TEXT)")
12 con.close()
13
14 app = Flask(__name__)
15 app.secret_key = '/sd_5#y2L"F4Q8z\n\xec)/'
16
17 app.config['UPLOAD_FOLDER']="static/images"
18
19 EMAIL=''
20
21 @app.route('/')
22 def index():
23     return render_template('signup.html')
24
25 @app.route('/home')
26 def home():
27     login=False
28     if 'usernameid' and 'passwordid' in session:
29         login=True
30     return render_template('index.html', login=login)
31
32 @app.route('/signup', methods=['POST','GET'])
33 @cross_origin()
34 def signup():
```

```

72     EMAIL=email
73     sql="SELECT * FROM authentication WHERE email=? AND password=?"
74     stmt=ibm_db.prepare(conn,sql)
75     ibm_db.bind_param(stmt,1,email)
76     ibm_db.bind_param(stmt,2,password)
77     ibm_db.execute(stmt)
78     account=ibm_db.fetch_assoc(stmt)
79     if account:
80         #session['loggedin'] = True
81         #session['email'] = email
82         #return render_template('home.html')
83         return redirect(url_for('home'))
84     else:
85         error = "Invalid email/password"
86         return redirect(url_for('login'))
87     elif request.method=="GET":
88         return render_template('login.html')
89
90 @app.route('/forgot', methods=['POST','GET'])
91 def forgot():
92     if request.method=="POST":
93         email=request.form['email']
94         remail=email
95         secret=request.form['secret']
96         sql="SELECT * FROM authentication WHERE email=? AND secret=?"
97         stmt=ibm_db.prepare(conn,sql)
98         ibm_db.bind_param(stmt,1,email)
99         ibm_db.bind_param(stmt,2,secret)
100        ibm_db.execute(stmt)
101        account=ibm_db.fetch_assoc(stmt)
102        if account:
103            return redirect(url_for('reset'))
104        else:
105            return redirect(url_for('forgot'))

```

```

p.py - D:\BM PROJECT\sprint1  app.py - C:\bm project - final
def post():
    con = sqlite3.connect("myimage.db")
    con.row_factory = sqlite3.Row
    cur = con.cursor()
    cur.execute("select * from image")
    data = cur.fetchall()
    con.close()

    if request.method=="POST":
        upload_image=request.files['upload_image']

        if upload_image.filename!='':
            filepath=os.path.join(app.config['UPLOAD_FOLDER'],upload_image.filename)
            upload_image.save(filepath)
            con=sqlite3.connect("myimage.db")
            cur=con.cursor()
            cur.execute("insert into image(img)values(?)",(upload_image.filename,))
            con.commit()
            flash("File Upload Successfully","success")

            con = sqlite3.connect("myimage.db")
            con.row_factory=sqlite3.Row
            cur=con.cursor()
            cur.execute("select * from image")
            data=cur.fetchall()
            con.close()
            return render_template("post.html",data=data)
    return render_template("post.html",data=data)

@app.route('/delete_record/<string:id>')
def delete_record(id):
    try:
        con=sqlite3.connect("myimage.db")
        cur=con.cursor()
        cur.execute("delete from image where id=?" [id])

```

```
index.html — Assignments\assignment 3\templates × index.html — github\...\templates × sample.html × app.py ×
1  {% extends 'base.html' %}
2
3  {% block head %}
4  <title>Home - Instant job</title>
5  <link rel="stylesheet" href="static/css/style.css">
6  <link rel="stylesheet" href="static/css/index.css">
7  {% endblock %}
8
9
10 {% block content %}
11
12
13
14 <div class="topbar">
15
16     <div class="topbarLeft">
17         <h2 class="topbarHead">INSTANT JOB</h2>
18     </div>
19
20     <div class="topbarCenter">
21         <ul class="topbarList">
22             <li class="topbarListItem"><a class="topbarListItem" href="contacts">Contacts</a></li>
23             <li class="topbarListItem"><a class="topbarListItem" href="login">Log in</a></li>
24             <li class="topbarListItem"><a class="topbarListItem" href="signup">Sign up</a></li>
25         </ul>
26     </div>
27
28 </div>
29
30 <div class="header">
31
32     <div class="headerImg">
33         <h3 class="headerTitle">WELCOME TO INSTANT JOB</h3>
34
35
```

```
app.py — D:\IBM PROJECT\sprint1 × app.py — C:\...\ibm project - final ×
1  import ibm_db
2  from flask import Flask, flash, redirect, render_template, request, url_for
3  from flask_cors import CORS, cross_origin
4  from flask import session
5  import sqlite3
6  import os
7
8  conn=ibm_db.connect("DATABASE=bludb;HOSTNAME=b1bc1829-6f45-4cd4-bef4-10cf081900bf.clogj3sd0tgu0lqde00.databases.appdomain.cloud;PORT=32304;")
9
10 con=sqlite3.connect("myimage.db")
11 con.execute("create table if not exists image(pid integer primary key,img TEXT)")
12 con.close()
13
14 app = Flask(__name__)
15 app.secret_key = '/sd_5#y2L"F4Q8z\n\xec]/'
16
17 app.config['UPLOAD_FOLDER']="static\images"
18
19 EMAIL=''
20
21 @app.route('/')
22 def index():
23     return render_template('signup.html')
24
25 @app.route('/home')
26 def home():
27     login=False
28     if 'usernameid' and 'passwordid' in session:
29         login=True
30     return render_template('index.html', login=login)
31
32 @app.route('/signup', methods=['POST', 'GET'])
33 @cross_origin()
34 def signup():
```

## 7.2 Feature 2

```
app.py — D:\BM PROJECT\sprint1 x app.py — C:\bm project - final x forgot.html x jobApplication.html x
118 <input class="shadow form-control" type="text" id="curcityid" name="curcity" required><br><br>
119 <label for="curstateid">State:</label>
120 <input class="shadow form-control" type="text" id="curstateid" name="curstate" required><br><br>
121 <label for="curcntryid">Country:</label>
122 <input class="shadow form-control" type="text" id="curcntryid" name="curcntryid" required><br><br>
123 </fieldset><br>
124
125 <fieldset>
126 <legend>QUALIFICATIONS</legend><br>
127 <table>
128 <tr>
129 <td align="center"><b>Sl.No.</b></td>
130 <td align="center"><b>Degree</b></td>
131 <td align="center"><b>Board</b></td>
132 <td align="center"><b>Percentage</b></td>
133 <td align="center"><b>Year of Passing</b></td>
134 </tr>
135
136 <tr>
137 <td>1</td>
138 <td>Class X</td>
139 <td><input type="text" name="Xboard" required></td>
140 <td><input type="text" name="XPercent" required></td>
141 <td><input type="number" name="XYOP" required></td>
142 </tr>
143
144 <tr>
145 <td>2</td>
146 <td>Class XII</td>
147 <td><input type="text" name="XIIBoard" required></td>
148 <td><input type="text" name="XIIPercent" required></td>
149 <td><input type="number" name="XIYOP" required></td>
150 </tr>
151
152 <tr>
```

Line 133, Column 83

Tab Size: 4

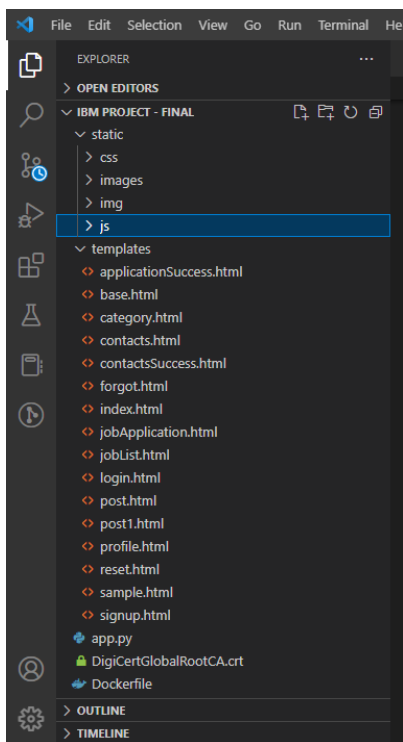
HTM

```
app.py — D:\BM PROJECT\sprint1  ×  app.py — C:\bm project - final  ×  forgot.html  ×  jobApplication.html  ×  jobList.html  ×
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="utf-8">
6      <title>Instant job</title>
7      <meta content="width=device-width, initial-scale=1.0" name="viewport">
8      <meta content="" name="keywords">
9      <meta content="" name="description">
10
11      <!-- Favicon -->
12      <link href="img/favicon.ico" rel="icon">
13
14      <!-- Google Web Fonts -->
15      <link rel="preconnect" href="https://fonts.googleapis.com">
16      <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
17      <link href="https://fonts.googleapis.com/css2?family=Heebo:wght@400;500;600&family=Inter:wght@700;800&display=swap" rel="stylesheet"
18      >
19
20      <!-- Icon Font Stylesheet -->
21      <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.10.0/css/all.min.css" rel="stylesheet">
22      <link href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.4.1/font/bootstrap-icons.css" rel="stylesheet">
23
24      <!-- Libraries Stylesheet -->
25      <link href="lib/animate/animate.min.css" rel="stylesheet">
26      <link href="lib/owlcarousel/assets/owl.carousel.min.css" rel="stylesheet">
27
28      <!-- Customized Bootstrap Stylesheet -->
29      <link href="css/bootstrap.min.css" rel="stylesheet">
30
31
32      <link rel="stylesheet" href="static/css/style.css">
33      <link rel="stylesheet" href="static/css/index.css">
34  </head>
```

```
app.py — D:\IBM PROJECT\sprint1 x app.py — C:\... \ibm project - final x forgot.html x jobApplication.html x jobList.html x reset.html x
1 [% extends 'base.html' %]
2
3 {% block head %}
4 <title>Forgot - Instant job</title>
5 <link rel="stylesheet" href="static/css/style.css">
6 <link rel="stylesheet" href="static/css/index.css">
7 {% endblock %}
8
9
10 {% block content %}
11
12 <div class="topbar">
13
14 <div class="topbarLeft">
15 <h2 class="topbarHead">INSTANT JOB</h2>
16 </div>
17
18 <div class="topbarCenter">
19 <ul class="topbarList">
20 <li class="topbarListItem"><a class="topbarListItem" href="signup">Sign up</a></li>
21 <li class="topbarListItem"><a class="topbarListItem" href="login">Log in</a></li>
22
23 </ul>
24 </div>
25 </div>
26
27 <section class="py-4 py-md-5 mt-5">
28 <div class="container py-md-5">
29 <div class="row d-flex align-items-center">
30 <div class="col-md-5 col-xl-4 text-center text-md-start">
31 <h2 class="display-6 mb-4">Reset your password</h2>
32 <p class="text-muted">Enter email and new password.</p>
33 <form method="post">
34 <div class="mb-3"><input class="shadow form-control" type="email" name="email" placeholder="Email"></div>
35 <div class="mb-3"><input class="shadow form-control" type="text" name="password" placeholder="Password"></div>
```

## 8 TESTING

### 8.1 Test Cases



## 8.2 User Acceptance Testing

The screenshot shows a database management tool interface. At the top, there are tabs for 'Load Data', 'Load History', 'Tables', 'Views', 'Indexes', 'Aliases', 'MQTs', 'Sequences', and 'Application objects'. The 'Tables' tab is selected. Below the tabs, there is a search bar with the text 'Find schemas or tables' and a 'Refresh' button. On the left side, there is a sidebar with icons for 'Schemas', 'SQL', and 'Tables'. The main area is divided into two panels. The left panel, titled 'Tables', shows a list of tables with columns 'Name', 'Schema', and 'Properties'. The table 'SIGNUP' is selected, and its details are shown in the right panel, titled 'Table definition'. The 'Table definition' panel shows the table's structure with columns 'NAME', 'EMAIL', and 'PASSWORD', all of type 'VARCHAR' and length 32. The table has approximately 3 rows (32.0 KB) and was updated on 2022-11-11 06:32:36. A 'View data' button is located at the bottom of the 'Table definition' panel.

Name	Data type	Nullable	Length	Scale
NAME	VARCHAR	Y	32	0
EMAIL	VARCHAR	Y	32	0
PASSWORD	VARCHAR	Y	32	0

## 9 RESULTS

### 9.1 Performance Metrics

The screenshot shows a terminal window with a Flask application running. The output shows the application starting on http://127.0.0.1:5000. The application is serving static files from the 'static' directory. The output shows several GET requests for static files, all returning 200 status codes. The application is running on a development server, and the output shows the Flask app is serving on http://127.0.0.1:5000. The output shows the Flask app is serving on http://127.0.0.1:5000. The output shows the Flask app is serving on http://127.0.0.1:5000.

```
-bef4-10cf081900bf.clogj3sd0tgtu0lqde00.databases.appdomain.cloud;P
ORT=32304;Security=SSL;SSLServerCertificate=DigiCertGlobalRootCA.cr
t;UID=cwy91974;PWD=aj53b8isyFaXlrZy;","")
Exception: [IBM][CLI Driver] SQL1336N The remote host "b1bc1829-6f
45-4cd4-bef4-10cf081900bf.clogj3sd0tgtu0lqde00.databases.ap" was no
t found. SQLSTATE=08001 SQLCODE=-1336
PS C:\Users\dasva\OneDrive\Desktop\ibm project - final> python -u "
c:\Users\dasva\OneDrive\Desktop\ibm project - final\app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a productio
n deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 143-483-106
127.0.0.1 - - [16/Nov/2022 11:50:42] "GET /home HTTP/1.1" 200 -
127.0.0.1 - - [16/Nov/2022 11:50:43] "GET /static/js/bootstrap.min.
js HTTP/1.1" 404 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/css/style.css HTT
P/1.1" 304 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/css/index.css HTT
P/1.1" 304 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/js/script.min.js
HTTP/1.1" 404 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/js/index.js HTTP/
1.1" 304 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/js/chat.js HTTP/1
.1" 304 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/img/man3.jpg HTTP
/1.1" 304 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/img/man2.jpg HTTP
/1.1" 304 -
127.0.0.1 - - [16/Nov/2022 11:50:44] "GET /static/css/bg2.jpg HTTP/
1.1" 304 -
```

## 9.2 Output

← → ↻ ⓘ 127.0.0.1:5000/signup

**INSTANT JOB** Log in

### Sign Up

Name

Email

Password

Create account

Already have an account? [Log in](#) →

127.0.0.1:5000/login



[Forgot your password?](#)




← → ↻ ⓘ 127.0.0.1:5000/jobList

INSTANT JOB

Home Explore Contact-us Log-out

Job openings


Featured Full Time Part Time



Software Engineer  
Accenture  
Chennai Full Time 4.6 lpa

Apply Now

Date Line: 01 Jan, 2022



Marketing Manager  
Amazon  
Bangalore Full Time 13 lpa

Apply Now

Date Line: 01 Jun, 2023

← → ↻ ⓘ 127.0.0.1:5000/home

INSTANT JOB

Jobs Explore Contact-us Log-out

View the latest job openings

Job openings

Explore jobs by various roles

Explore roles

**Instant Job**  
The most powerful platform  
for job seekers as well as  
recruiter

**Discover**  
Web design  
Development  
Hosting

**About**  
Company  
Team  
Legacy

**Careers**  
Job openings  
Employee success  
Benefits

← → ↻ ⓘ 127.0.0.1:5000/jobApplication

🔗 ☆ ⚙ □ 🌐

INSTANT JOB

ContactLog inSign up

## APPLICATION FORM

### PERSONAL DETAILS

First Name:

Last Name:

Email (Enter same as account email id):

Phone number:

## 10 ADVANTAGES & DISADVANTAGES

### ADVANTAGES

- When recruiting externally, hiring teams find candidates, evaluate them and, if all goes well, persuade them to join their company. All of which takes time.
- Everyone needs some time to adjust to a new role, but internal hires are quicker to onboard than external hires.
- May be familiar with people in their new team, especially in smaller businesses.
- Know how your company operates and most of your policies and practices.

### DISADVANTAGES

- Employees who were considered for a role could feel resentful if a colleague or external candidate is eventually hired.
- While your company may have a lot of qualified candidates for specific positions, this isn't

necessarily true for every open role.

## **11 CONCLUSION**

By the end of this project we will

- know fundamental concepts and techniques of recommender system.
- gain a broad understanding of databases and cloud.
- know how to build a web application using the Flask framework.
- know how to build chatbot.
- know how to containerize the application.

## **12 FUTURE SCOPE**

- AI is revolutionizing the recommender systems.
- The popularity of LinkedIn and Google for jobs has proved that there is a future for job boards if effectively managed to provide solutions
- Right pricing strategies for online recruitment advertising are essential to get an effective response.
- Recruiters and job seekers are experiencing an entirely automated process of searching and connecting. All job boards should be perfectly indexed, highly responsive, and exhaustive in job descriptions to establish their credibility and reliability.

## **13 APPENDIX**

GitHub - <https://github.com/IBM-EPBL/IBM-Project-3658-1658588080>