# SKILL/JOB RECOMMENDER APPLICATION

## A NAALAIYATHIRAN PROJECT REPORT

Submitted by,


| | |
|---|---|
| Vijay.R | 610519104111 |
| Navrang.K.P | 610519104070 |
| Vishnu.R | 610519104115 |
| Sudhagar.P | 610519104100 |


**TEAM ID: PNT2022TMID29859**

# TABLE OF CONTENT

# CHAPTER 1
# 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 jobsearch API to get the current job openings in the market which will fetch the data directly from the webpage.

## 1.2. PURPOSE OF THIS PROJECT

- Students/Job seekers can get the desired job based on their skill set.
- We can share the profiles with companies and generate the revenue by providing them best profiles.
- We can provide the application for job seekers in a subscription based.
- Integrating a Chatbot will helpful for the students to interact with this web application to seek a particular job

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 EXISTING SOLUTION

- https://www.hirist.com/login
- https://www.linkedin.com

## 2.2 REFERENCES

### 1. "Students / Job seekers find their desired job based on their Skillset"

**Reference link**: https://www.researchgate.net/publication/272802616_A_survey_of_job_recommender_systems

**Description:**

The Internet-based recruiting platforms become a primary recruitment channel in most companies. The recommender system technology aims to help users in finding items that match their personnel interests. This article will present a survey of e-recruiting process and existing recommendation approaches for building personalized recommender systems for candidates/job matching.

### 2. "Integrating Intelligent CHATBOT for Job recommendation application"

**Reference link:** https://www.researchgate.net/publication/360820692_Intelligent_Chatbot

**Description:**

A Chatbot is a software application that replaces a live human agent to conduct a conversation via text or text to speech. In this system, we demonstrate a chatbot that uses Artificial Intelligence to produce dynamic responses to online client enquiries. This web-based platform provides a vast intelligent base that can help humans to solve problems. The Chatbot recognizes the user's context, which prompts an intended response. Its objective is to reduce human dependency in every organization and reduce the need for different systems for different processes.

## 3. “A Study of LinkedIn as an Employment Tool for Job Seeker & Recruiter”

**Reference link:** Journal homepage: www.ijrpr.com ISSN 2582-7421 **Description:**

LinkedIn has become one of the most known social networking portals in terms of global professional connections, networking, job postings, hiring and much more in relevance to employment opportunities.

This research was an attempt to identify the utility of Linked in on selection and recruitment. Also, this study has taken the employers’ and the prospective candidates for job and employees’ perspective, including factors such as recruitment, selection, job opportunities, internal official communication on Linked-in, professional networking, ease of access, less expensive communication tool etc.

## 4. "Cloud storage and sharing services"

**Reference link:** https://www.ijresm.com/

**Description:**

To create a web application that sends files from one email to another email using the SMTP protocol, which is handled in a server-based application. The main advantage of the project in this paper is that it provides a safe, reliable, and excellent tool for sharing files in any format. Also, it has infinite scaling capabilities. With a bit of tweak in the code, it can be scaled to handle heavy file loads.

The Cloud-based file sharing approach is proposed to provide the following services for external data confidentiality, secure data sharing within the group, protect data from unauthorized access of officials within the group and provide time and number of file access to users. Whenever information sharing among a bunch arise the file owner sends the user uploads the file on the application and then shares it using the send API. This creates a safe medium of sharing of files and user in control of the data in the whole process of sharing the files.

## 2.3 . PROBLEM STATEMENT DEFINITION

Dealing with the enormous amount of recruiting information on the Internet, a job seeker always spends hours to find useful ones. Many times, people who lack 3 industry knowledge are unclear about what exactly they need to learn in order to get a suitable job for them. We address the problem of recommending suitable jobs to people who are seeking a new job. Job recommender technology aims to help job seekers in finding jobs that match their skills. The Internet caused a substantial impact on the recruitment process through the creation of e-recruiting platforms that become a primary recruitment channel in most companies. While companies established job positions on these portals, job-seeker uses them to publish their profiles. E-recruitment platforms accomplished clear advantages for both recruiters and job-seekers by reducing the recruitment time and advertisement cost. Recommender system technology aims to help users in finding items that match their preferences; it has a successful usage in a widerange of applications to deal with problems related to information overload efficiently. In order to improve the e-recruiting functionality, many recommender system approaches have been proposed. This paper will analyze e-recruiting process and related issues for building personalized recommender systems of candidates/job matching.

# CHAPTER 3

# IDEATION & PROPOSED SOLUTION

## 3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users. Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.


Reference: https://www.mural.co/templates/empathy-map-canvas

## 3.2. IDEATION

- E-recruitment platforms decrease the recruitment time and advertisement cost, they suffer from an inappropriateness of traditional information retrieval techniques like the Boolean search methods. Consequently, a vast amount of candidates missed the

opportunity of recruiting. The recommender system technology aims to help user in finding items that match their personnel interests; it has a successful usage in ecommerce applications to deal with problems related to information overload efficiently.

- Job recommendation application with intelligence of chatbot. In this system, we demonstrate a chatbot that uses Artificial Intelligence to produce dynamic responses to online client enquiries. This web-based platform provides a vast intelligent base that can help humans to solve problems. The chatbot recognizes the user's context, which prompts an intended response.

  Because this is a dynamic response, the user's desired response will be generated. This also uses a machine-learning algorithm to learn the chatbot by experiencing various requests and responses. Chatbots come to use in numerous fields of our daily life. Because AI enhances the human touch in every communication, chatbot are becoming increasingly robust. It triggers accurate responses after understanding a user's query. Its objective is to reduce human dependency in every organization and reduce the need for different systems for different processes.

- Job seekers struggling to get the desired job for skills they have. we are proposing an application which will help the students to give Suggestions on the jobs based the skills. In this application freshers or skilled person can sign up and find the jobs by using either the search option or they can directly interact with the chatbot and get their dream job. In this application freshers or skilled person can sign up and find the jobs by using either the search option or they can directly interact with the chatbot and get their dream job. 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.

## 3.3. PROPOSED SOLUTION

| S.NO | PARAMETER | DESCRIPTION |
|---|---|---|
| 1 | Problem Statement (Problem to be solved) | 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. |
| 2 | Idea / Solution description | 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. |
| 3 | Novelty / Uniqueness | A chatbot can easily save time and effort. These chatbot communicate with humans in a very natural form to resolve their queries and to assist them as required. |

| | | |
|---|---|---|
| 4 | Social Impact / Customer Satisfaction | The user allowed to choose the required job as per his/her skill level. It helps user to make right decision to choose their required field job |

| | | |
|---|---|---|
| 5 | Business Model (Revenue Model) | We can provide the application in a subscription based. It will help the people to track their expenses and also alerts when you exceed the limit of your budget. |
| 6 | Scalability of the Solution | One important application of recommendation system in Job Recruitment; in which candidates are elected by using online job recruitment portal based on their profile and job history and behavior components. |

# 3.4. PROBLEM SOLUTION FIT

**Project Title:** Skill/Job recommended application

**Project Design Phase – 1**
Problem Solution Fit

**Team Id:** PNT2022TMID29859

Define CS, fit into

**1. CUSTOMER SEGMENT(S)** CS

Who is your customer?

The main customers for our project are :

- Person who are seeking for employment
- Person that recruit job candidates

**6. CUSTOMER** 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.

- Concern about misuse of personal information
- Worry about unreliable connections
- Potential scam
- Time consuming
- Needs understanding to use the application

**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

- Text processing and recommendation method
- Content-based filtering
- Collaborative filtering
- Graph-based filtering

Explore AS,

Focus on J&P, tap into BE, understand

**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.

- Confusion in choosing a right job
- Similar job alerts for frequent times
- Many of the jobs are not real
- The companies listed do not give their actual structure

**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.

- Giving incorrect details in the profile page
- Network problem
- User post false credentials
- Some job portals want payment in advance of the job starting
- The company and the job openings should be verified

**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)

- User-friendly
- Chat support
- Providing the actual infrastructure of the company
- Cheating during online recruitment process

Focus on J&P, tap into BE, understand

Identify strong TR & EM

**3. TRIGGERS** TR

- The user gets the job alert
- Job description revels the necessary data

**4. EMOTIONS: BEFORE / AFTER** EM

Before: Looking for a dream job
After: Feel comfortable they can stay connected and employed

**10. YOUR SOLUTION** SL

To develop an end-to-end web application which in default have a lot of current job openings through job search API out of which appropriate job will be recommended based on user skill set. At the same time students can develop their skills side by side with various courses and webinars offered by reputed organization. In addition to this a smart chat bot will be available for 24*7 which can help users in finding the right job.

**8. CHANNELS of BEHAVIOUR** CH

8.1 ONLINE

Users have to upload their resumes and fill up the essential details such as name, education, skills, location, and experience

8.2 OFFLINE

User can view the job description from their alerts

Extract online & offline CH of BE

# CHAPTER 4

# REQUIREMENT ANALYSIS

## 4.1 FUNCTIONAL REQUIREMENT:

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form<br>Registration through Gmail<br>Registration through LinkedIN |
| FR-2 | User Login | Confirmation via Email<br>Confirmation via OTP |
| FR-3 | Get Job Recommendation through search API | We can use your job search API to get the current job opening in the market which will fetch the directly from the web page |
| FR-4 | Job Alerts | Various modules triggers job alert. A notification is sent to the user whenever there is a job opening based on the user's skillset. |
| FR-5 | ChatBot to solve user query | A chatbot can help answer FAQ's about applying, benefits or next steps in the application process. |

## 4.2 NON-FUNCTIONAL REQUIREMENT:

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | Usability | In case of job recommender application, Usability refers to the quality of a user's experience when interacting with the web UI .Usability is about effectiveness, efficiency and the overall satisfaction of the user. Usability describes the level of ease with which a system allows a user to get to that Goal. |
| NFR-2 | Security | Authentication:<br>Authentication is the act of determining the identity of a user. For example, User id and Password, Biometric Identification.<br>Authorization:<br>Authorization is the act of determining the level of access that an authorized user has to behaviour and data (use of firewalls). |
| NFR-3 | Reliability | In a perfect world, a reliable piece of software is completely defect free, does not create downtime, and performs correctly in every scenario. |
| NFR-4 | Performance | In Job Recommender Application performance indicates how the app is functioning and how responsive the app is to the end-user. There are tools available to measure Application Performance and enable app developers to detect and diagnose complex application performance problems to maintain the expected level of service to end-users. |

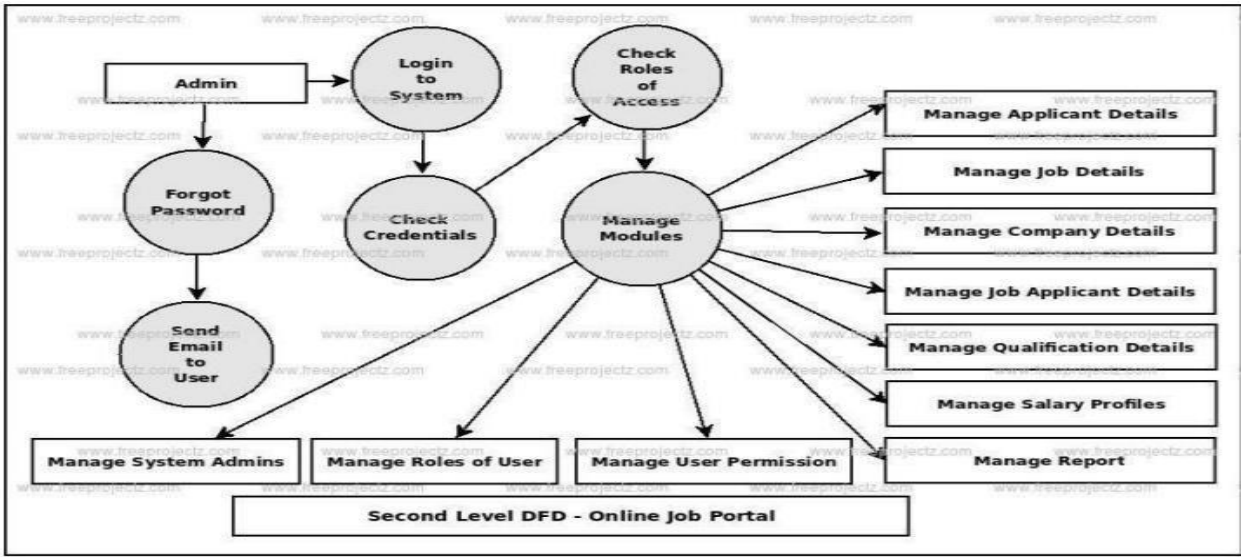
| | | |
|---|---|---|
| NFR-5 | Availability | High availability systems are important in many industries because they help ensure systems function correctly for a continuous period. Technology and networks occasionally fail, such as when there's a power outage or a server error. In some industries, it's essential that the network remains functional at all times. |
| NFR-6 | Scalability | One important application of recommendation system in Job Recruitment; in which candidates are elected by using online job recruitment portal based on their profile and job history and behaviour components. |

# CHAPTER 5

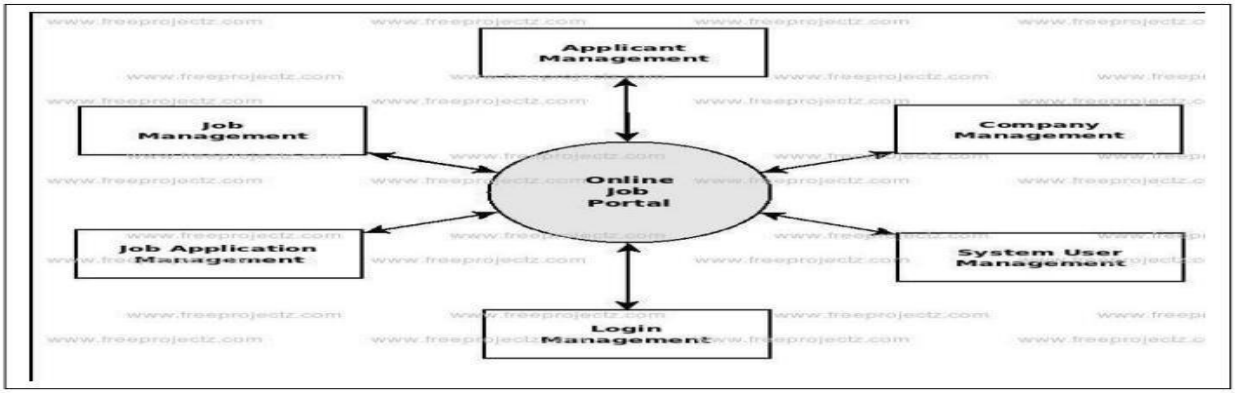
# PROJECT DESIGN

## 5.1 DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. It shows how data enters and leaves the system, what changes the information, and where data is stored.




**Example:** DFD Level 0 (online job portal)



## 5.2 USER STORIES

| User Type | Functional Requirement (Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|
| Customer (Mobile user) | | 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 online job web portal and jobs are recommended to the user skillsets. | I can register & access the dashboard with online web job portal Login | Low | Sprint-2 |
| | | USN-4 | As a user, I can register for the application through Gmail | I can register& access the dashboard | Medium | Sprint-1 |
| | Login | USN-5 | As a user, I can log into the application by entering email & password | I can receive confirmation email | High | Sprint-1 |
| | Dashboard | | As a user, I can register for the application shown in the online job web portal which is suited for my skillsets. | | High | |
| Customer (web user) | | | As a job seeker, user, freshers or experienced candidates.<br>They want job recommendation email and known the recently updated job information through email or message.<br>So that they were aware about the jobs based on their skillsets. | | | |
| Customer Care Executive | | | As a user, I want to record all the candidate application details based on the skillsets of the candidates like freshers, jobseekers and experienced candidate at the end of the given deadline. | | | |
| Administrator | | | As a admin, I want my requisition automatically populated with current job opportunities. | | | |

## 5.3 SOLUTION ARCHITECTURE

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

- Job seekers struggling to get the desired jobs for the skills they have. They tend to miss out on job openings because there is a many opportunities to get their desire job that list millions of jobs which are generally not relevant at all to the users.

- In job recommender application job seekers can showcase their skills through their profiles. The user and their information are stored in the Database. User can find the jobs by using the search option or they can directly interact with the Chabot and get recommendation on Jobs based on their skillset.

- A notification is send to the user through E-mail, whenever there is a job opening based on the user's skillset.

- Job recommendation facilitates that experience, instead of having to scroll through a lot of job openings. We can easily get a job recommendation by using search API or Chabot. User can find easily their desire job through Email.

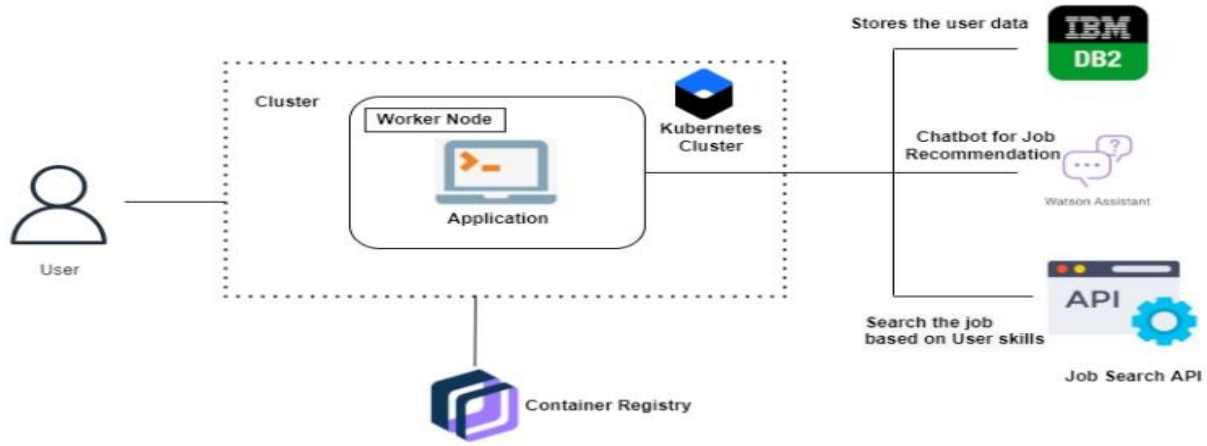



*Figure 1: Architecture and data flow of the Skill/Job recommender application*

**Reference:**

https://www.researchgate.net/publication/325697854_Job_Recommendation_based_on_Job_Seeker_Skills_An_Empirical_Study

# CHAPTER 6

# PROJECT PLANNING & SCHEDULING

## 6.1 SPRINT PLANNING & ESTIMATION:

| Sprint | Functional Requirement (Epic) | User Story Number | User Story / Task | Story Points | 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. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-1 | | USN-2 | As a user, I will receive a confirmation email once I have registered for the application. | 1 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-1 | | USN-3 | As a user, I can register for the application through Gmail. | 2 | Medium | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-1 | Login | USN-4 | As a user, I can log into the application by entering my email & password. | 1 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-1 | Dashboard | USN-5 | As a user, I can access the website in a second. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-1 | Dashboard | USN-6 | As a user, If I Log in correctly, I can view my dashboard and I can navigate to any pages which are already listed there. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-2 | User Profile | USN-7 | As a user, I can view and update my details. | 2 | Medium | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-2 | Database | USN-8 | As a user, I can store my details and data in IBM Database. | 2 | Medium | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-2 | Cloud Storage | USN-9 | As a user, I can upload my photo, resume and much more in the website. | 1 | Medium | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-2 | Chatbot | USN-10 | As a user, I can ask the Chatbot about the latest job openings, which will help me and show the recent job openings based on my profile. | | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-2 | Identity-Aware | USN-11 | As a User, I can access my account by entering the correct login credentials and my user credentials are only displayed to me. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-3 | Learning Resource | USN-12 | As a user, I can learn the course and I will attain the skills which will be useful for developing my technical skills. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-3 | Docker | USN-13 | As a user, I can access the website in any device. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint-3 | Kubernetes | USN-14 | As a user, I can access the website in any device. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint 3 | Development in cloud | USN-15 | As a user, I can access the website in any device. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |
| Sprint 3 | Technical Support | USN-16 | As a user I can get customer care support | 1 | Medium | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint 4 | Unit Testing | USN-17 | As a user, I can access the website without any interruption. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |

| | | | | | | |
|---|---|---|---|---|---|---|
| Sprint-4 | Integration testing | USN-18 | As a user, I can access the website without any interruption. | 2 | High | Vijay R, Navrang K P, Vishnu K,Sudhagar P. |
| Sprint-4 | System testing | USN-19 | As a user, I can access the website without any interruption. | 2 | High | Vijay R, Navrang K P, Vishnu K, Sudhagar P. |

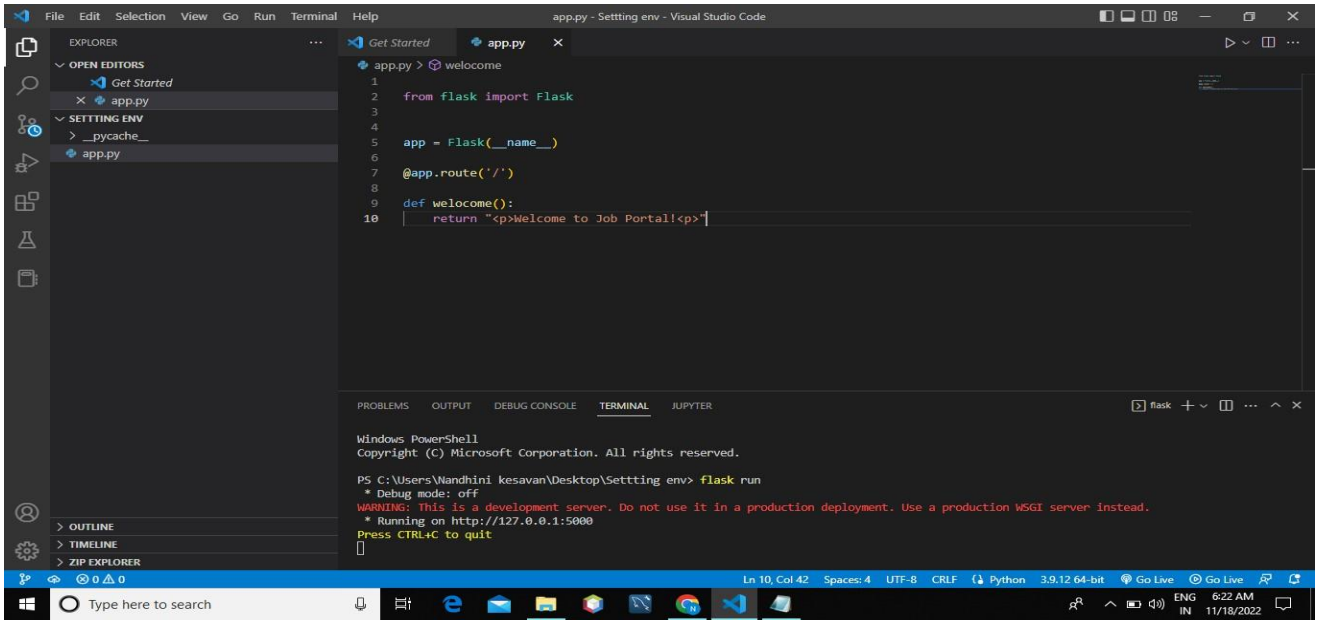
## 6.2 SPRINT DELIVERY SCHEDULE:

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

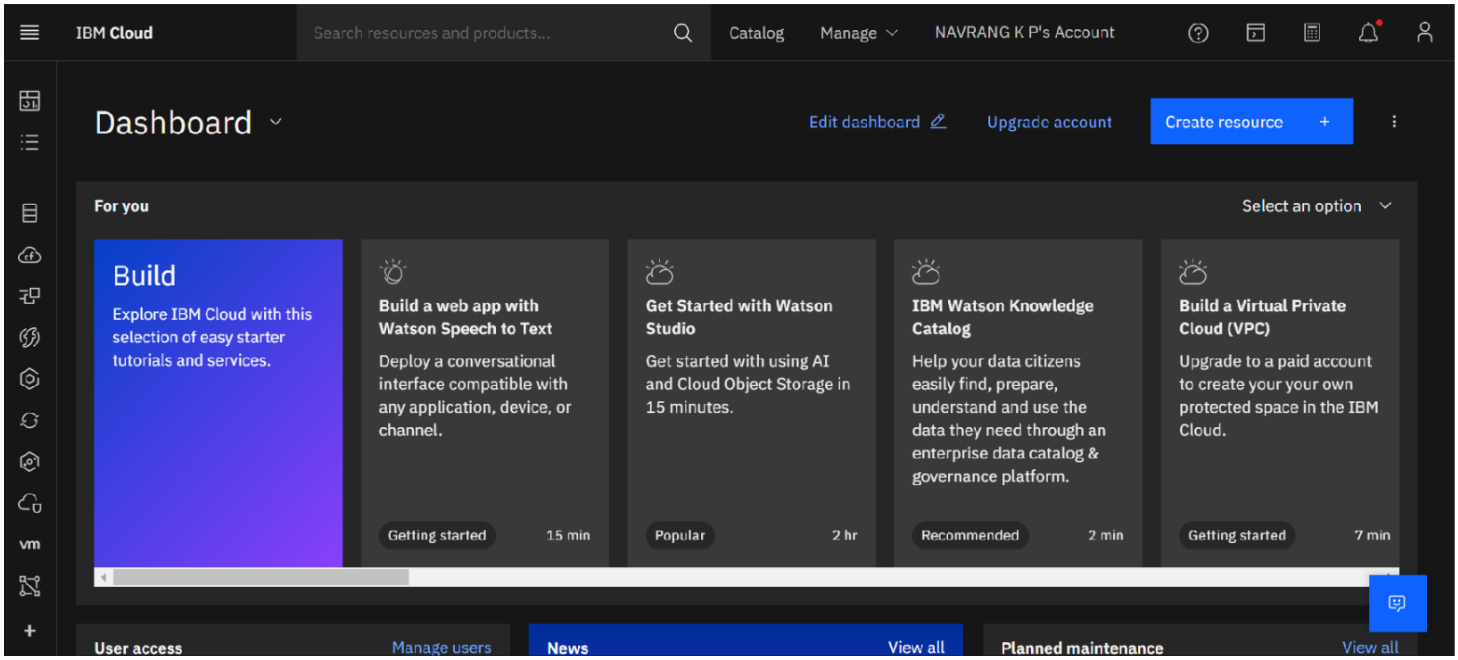
# CHAPTER 7

# SETTING UP AN ENVIRONMENT

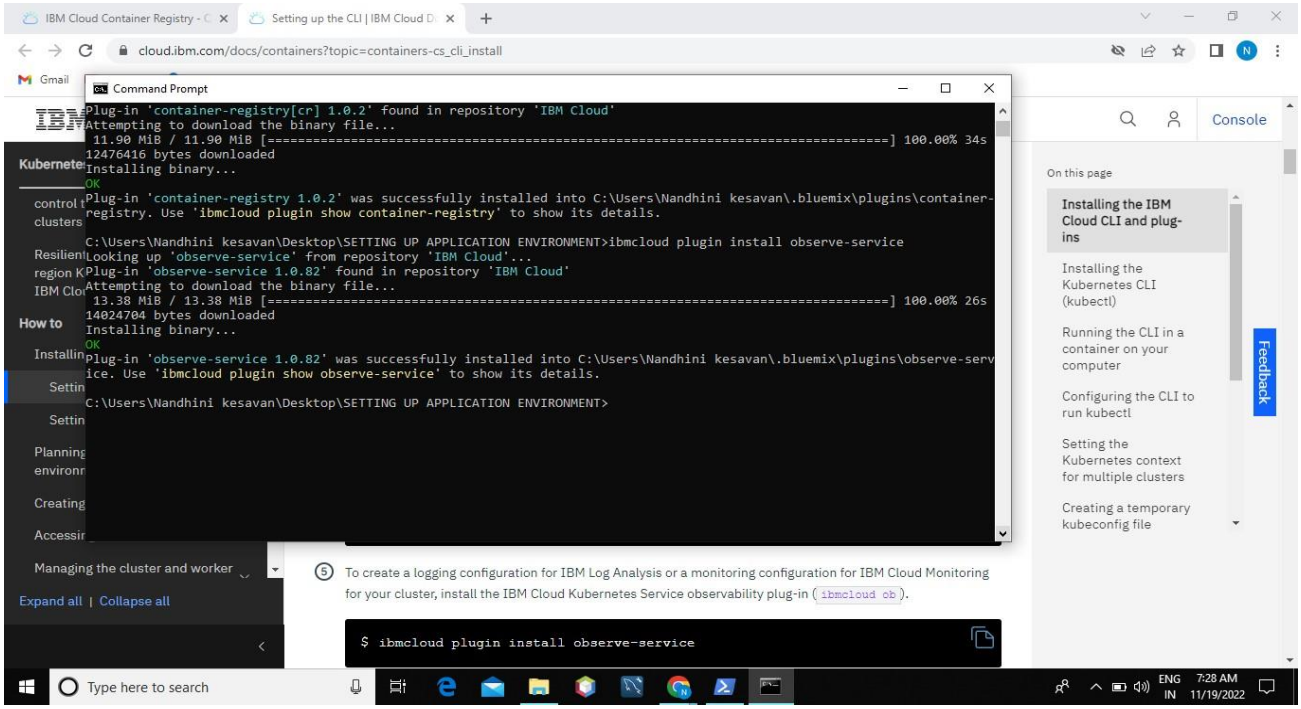
## 7.1. CREATE FLASK PROJECT



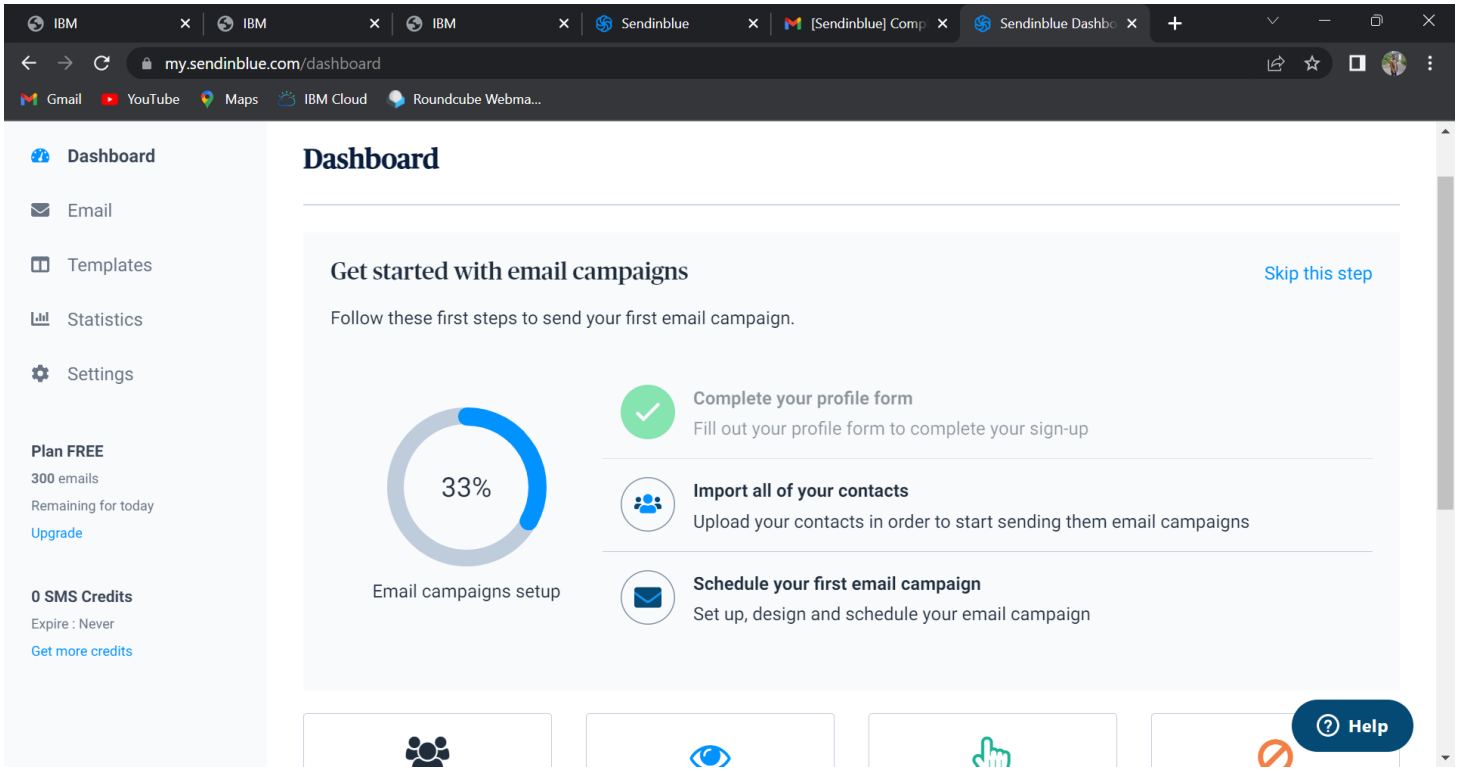

## 7.2. CREATE IBM CLOUD ACCOUNT



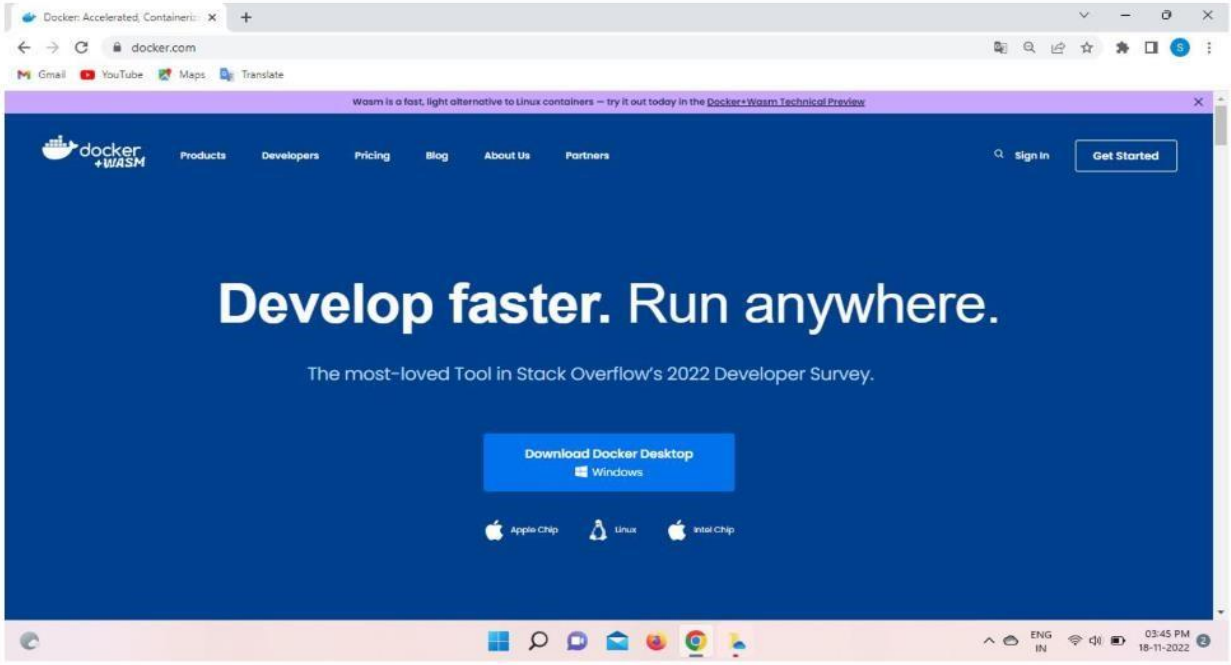
## 7.3. INSTALL IBM CLOUD CLI




## 7.4. CREATE AN ACCOUNT IN SENDGRID



## 7.5. DOCKER CLI INSTALLATION



# CHAPTER-8

# CODING & SOLUTIONING

## (Explain the features added in the project along with code)

## 8.1 FEATURE

## IBM WATSON ASSISTANT:

IBM Watson assistant uses artificial intelligence that understands customers in context to provide fast, consistent, and accurate answers across any application device or channel.

A job recommender application facilitates the experience , instead of having to scroll through a lot of job openings, We can use either chatbot or search API to know about your desired job.

In Chatbot people only need to answer a few questions and get the perfect recommendation within a minutes, if not seconds.

## 8.2 DATABASE CONNECTIVITY

```
from markupsafe import escape

import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLServerC
ertificate=DigiCertGlobalRootCA.crt;UID=csr80916;PWD=yUE8PuQT3mYrvp9N",'','')
print(conn)
print("Login sucessful")

app = Flask(__name__)
app.secret_key = '32y[wld,fnpsygfwfpwek2;]1[2'

@app.route('/')
def home():
    message = "TEAM ID : PNT2022TMID29835" +" "+ "BATCH ID : B1-1M3E "
    return render_template('index.html')

@app.route('/login', methods=['GET','POST'])
def login():
    return render_template('login.html')

@app.route('/register', methods = ['GET','POST'])
def register():
    return render_template('register.html')

@app.route('/studentdashboard', methods = ['GET','POST'])
def studentdashboard():
    return render_template('Stdash.html')

@app.route('/industrydashboard', methods = ['GET','POST'])
def industrydashboard():
    return render_template('Indusdash.html')

@app.route('/changepass', methods = ['GET','POST'])
def changepass():
    return render_template('changepass.html')

@app.route('/register_industry', methods=['GET', 'POST'])
def register_industry():
    if request.method == 'POST':
        name = request.form['Nm']
        email = request.form['email']
        phonenumber = request.form['PhNo']
        password = request.form['pass']
```

```
        sql = "SELECT * FROM industry 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)
```

# CHAPTER 9

# TESTING

## 9.1 TEST CASES

A test case is a set of rules or conditions to check if the system or one of its feature works in accordance to the requirement. It is a document with a set of details which includes, set of test data, expected results, actual results, environment information and soon.I have designed and executed a few test cases to check if the application meets the functional requirements.

Below are the test cases for the Job Search Portal web application.

| **TEST MODULE** | **TEST CASE** | **EXPECTED RESULT** | **TEST RESULT** |
|---|---|---|---|
| ADMIN | Provide valid login credentials | User successfully logged in and directed to the admin dashboard page | PASS |
| ADMIN | Enters invalid login credentials | Displays Error message | PASS |
| ADMIN | Upon successful login, click on the 'List of Employers' tab. | Displays the details of list of active employers registered with the application | PASS |
| EMPLOYER | Provide details for registration | Employer successfully registered with the application | PASS |
| EMPLOYER | Upon successful login, click on 'Post New Job' tab | Employer posts jobs with the required details | PASS |

| | | | |
|---|---|---|---|
| JOBSEEKER | Provide details for registration | Jobseeker successfully registered with the application | PASS |
| JOBSEEKER | Enters invalid login credentials | Error message displayed | PASS |
| JOBSEEKER | Upon successful login, click on 'Search Jobs' tab | Details of the active job postings are displayed. | PASS |

## 9.2. RESULTS

**Test Objective:** Navigation from splash screen to jobs screen

| **TEST CONDITION** | **INPUT SPECIFICATION** | **OUTPUT SPECIFICATION** | **PASS/FAIL** |
|---|---|---|---|
| The user currently on the splash screen | Users enters credentials and clicks on login button | Direct to jobs screen | PASS |

**Test Objective:** Navigation from jobs screen to job details screen

| **TEST CONDITION** | **INPUT SPECIFICATION** | **OUTPUT SPECIFICATION** | **PASS/FAIL** |
|---|---|---|---|
| The user currently on jobs screen | Users click on the view against a particular job | Direct to job details screen | PASS |

# CHAPTER 10

# ADVANTAGES &DISADVANTAGES

## 10.1. ADVANTAGES:

- **Getting Job Alerts:**

  A superior job portal provides standard job alerts whenever there are job openings for you. This way, you won't neglect a chance to be valid for your target job and bound to start your career. Also, you can discover more posts to increase better options.

- **Classified:**

  Once you register in a job portal, all your achievements, skills, and individual details will be reserved privately. This will be kept secure unless you let them split your details to your future employers. Also, job portals keep your job search narration classified which permits you to do the task surreptitiously.

- **More job opportunities:**

  Job portals offer a full collection of job choices from top companies. It means you have a superior chance of searching the job you want anywhere and however; you want it.

- **Resourceful:**

  With these portals, you can be relevant for a job effortlessly. Instead of going to your intention firm, you can submit your resume online. After submitting, you can relax, and wait for your future employers to take action.

## 10.2. DISADVANTAGES:

- **They have no screening.**

Anyone can sign up for a profile and apply for a job on an online job portal - no matter their credentials. With no official screening process in place, the website you're using can allow unlimited unqualified candidates to apply. Reviewing unqualified applications is an even further waste of your time.

- **They're ineffective for high-level positions.**

Because they have no screening process and allow companies to post for a wide variety of jobs, online job search websites tend to be better choices for entrylevel positions. Your high-level position won't get the attention or the kind of candidates you need

# CHAPTER-11

# CONCLUSION

We proposed a framework for job recommendation task. This frame work facilitates the understanding of job recommendation process as well as it allows the use of a variety of text processing and recommendation methods according to the preferences of the job recommender system designer. Moreover, we also contribute making publicly available a new dataset containing job seekers profiles and job vacancies. Future directions of our work will focus on performing a more exhaustive evaluation considering a greater amount of methods and data as well as a comprehensive evaluation of the impact of each professional skill of a job seeker on the received job recommendation.

# CHAPTER-12

# FUTURE SCOPE

- The job boards have to keep up with the shift in consumer base and market demands. There has been a significant change happening in the digital platform in terms of the content the generation consumes and demands due to differences in political, technological, and economic points of view. Therefore, the job boards must focus on offering value propositions that are quite different and relevant as well.

- 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. These features can be clubbed with technical upgrades like job tags supported with search engine optimizations and resume-matching criteria that are need of the hour.

- The job boards will be ready to face challenges from social media and professional networking sites. Job boards need to figure out what are the gaps and modify strategies that can make them more popular for job seekers and employers. The future generation relies more on digital networking tools to look for job searches. Easy accessibility to digital tools and techniques can help establish better connections which makes the job portals remain on the competitive edge.

# CHAPTER-13s

# APPENDIX

# SOUCE CODE (SAMPLE)

## 14.1 SOURCE CODE

### app.py

```
from flask import Flask, render_template,url_for,redirect,request,session
from werkzeug.utils import secure_filename
from importlib.resources import contents
from tkinter import S
from turtle import title
from flask import Flask,redirect, render_template, request, session, url_for,flash
from pyexpat import model
from sqlalchemy import PrimaryKeyConstraint
from werkzeug.utils import secure_filename

from markupsafe import escape

import ibm_db
conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ea286ace-86c7-4d5b-8580-
3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;PORT=31505;SECURITY=SSL;SSLServerC
ertificate=DigiCertGlobalRootCA.crt;UID=csr80916;PWD=yUE8PuQT3mYrvp9N",'','')
print(conn)
print("Login sucessful")

app = Flask(__name__)
app.secret_key = '32y[wld,fnpsygfwfpwek2;]1[2'

@app.route('/')
def home():
    message = "TEAM ID : PNT2022TMID29835" +" "+ "BATCH ID : B1-1M3E "
    return render_template('index.html')

@app.route('/login', methods=['GET','POST'])
def login():
    return render_template('login.html')

@app.route('/register', methods = ['GET','POST'])
def register():
    return render_template('register.html')
```

```
@app.route('/studentdashboard', methods = ['GET','POST'])
def studentdashboard():
    return render_template('Stdash.html')

@app.route('/industrydashboard', methods = ['GET','POST'])
def industrydashboard():
    return render_template('Indusdash.html')

@app.route('/changepass', methods = ['GET','POST'])
def changepass():
    return render_template('changepass.html')

@app.route('/register_industry', methods=['GET', 'POST'])
def register_industry():
    if request.method == 'POST':
        name = request.form['Nm']
        email = request.form['email']
        phonenumber = request.form['PhNo']
        password = request.form['pass']

        sql = "SELECT * FROM industry 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)

        if account:
            flash("Record Aldready found", "success")

        else:
            insert_sql = "insert into
industry(name,email,phonenumber,password)values(?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, phonenumber)
            ibm_db.bind_param(prep_stmt, 4, password)
            ibm_db.execute(prep_stmt)
            return redirect(url_for("login"))

@app.route('/register_student', methods=['GET', 'POST'])
def register_student():
    if request.method == 'POST':
        name = request.form["Nm"]
        email = request.form["email"]
        phonenumber = request.form['PhNo']
        password = request.form['pass']
```

```
        sql = "SELECT * FROM student1 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)

        if account:
            flash("Record Aldready found", "success")

        else:
            insert_sql = "insert into
student1(name,email,phonenumber,password)values(?,?,?,?)"
            prep_stmt = ibm_db.prepare(conn, insert_sql)
            ibm_db.bind_param(prep_stmt, 1, name)
            ibm_db.bind_param(prep_stmt, 2, email)
            ibm_db.bind_param(prep_stmt, 3, phonenumber)
            ibm_db.bind_param(prep_stmt, 4, password)
            ibm_db.execute(prep_stmt)
            return redirect(url_for('login'))

@app.route('/login_industry', methods=['GET', 'POST'])
def login_industry():
    if request.method == 'POST':
        mail = request.form['em']
        password = request.form['pass']
        print(id, password)
        sql = f"select * from industry where email='{escape(mail)}' and
password='{escape(password)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        data = ibm_db.fetch_both(stmt)

        if data:
            session["mail"] = escape(mail)
            session["password"] = escape(password)
            return redirect(url_for("industrydashboard"))

        else:
            return redirect(url_for("login",msg = "Account does not exits or invalid"))
    else:
        return "NOT WORKING"

@app.route('/login_student', methods=['GET', 'POST'])
def login_student():
    if request.method == 'POST':
        mail = request.form["em"]
        password = request.form["pass"]
```

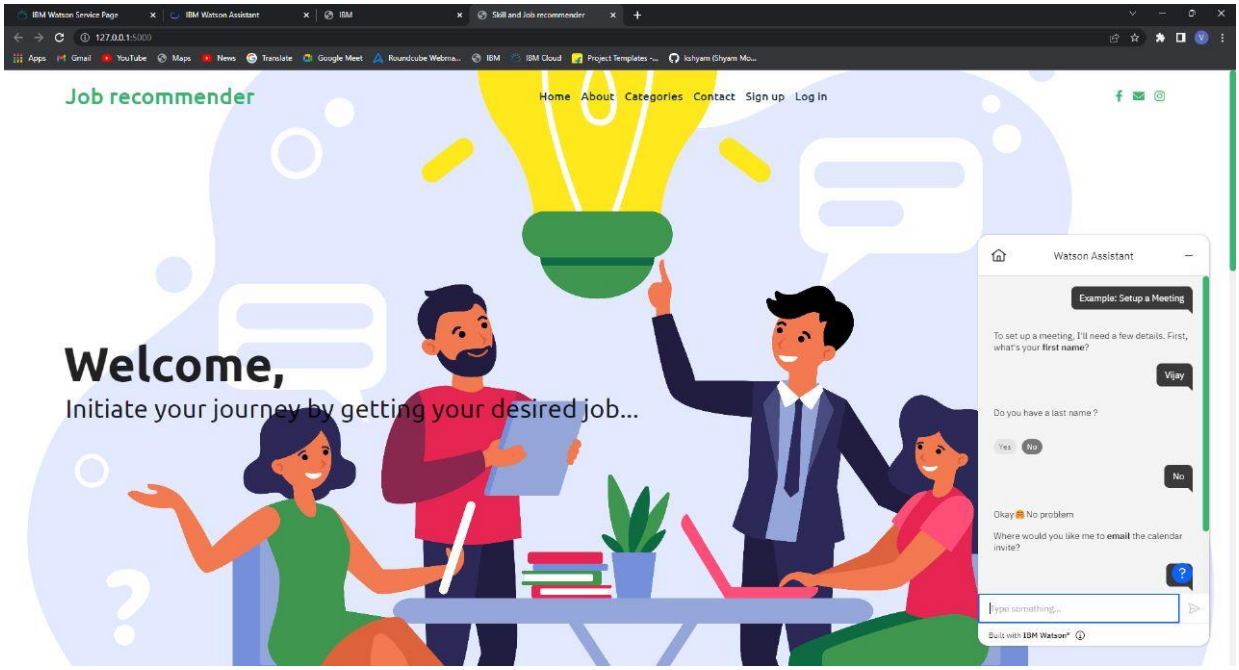
```
        sql = f"select * from student1 where email='{escape(mail)}' and
password='{escape(password)}'"
        stmt = ibm_db.exec_immediate(conn, sql)
        data = ibm_db.fetch_both(stmt)

        if data:
            session["em"] = escape(mail)
            session["password"] = escape(password)
            return redirect(url_for("studentdashboard"))

        else:
            return redirect(url_for("login",msg = "Account does not exits or invalid"))

    else:
        return "NOT WORKING"



if __name__ == "__main__":
    app.run(debug=True)
```

# APPENDIX OUTPUT

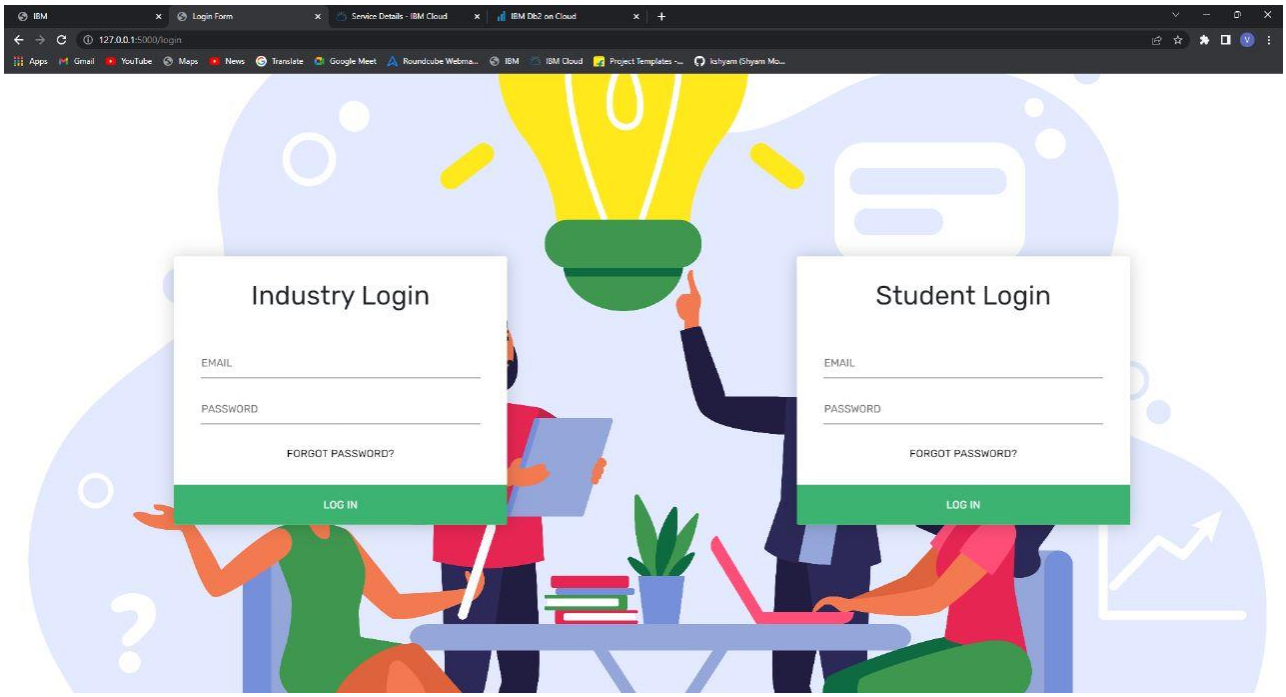
# (SCREENSHOT)

## USER INTERFACE:

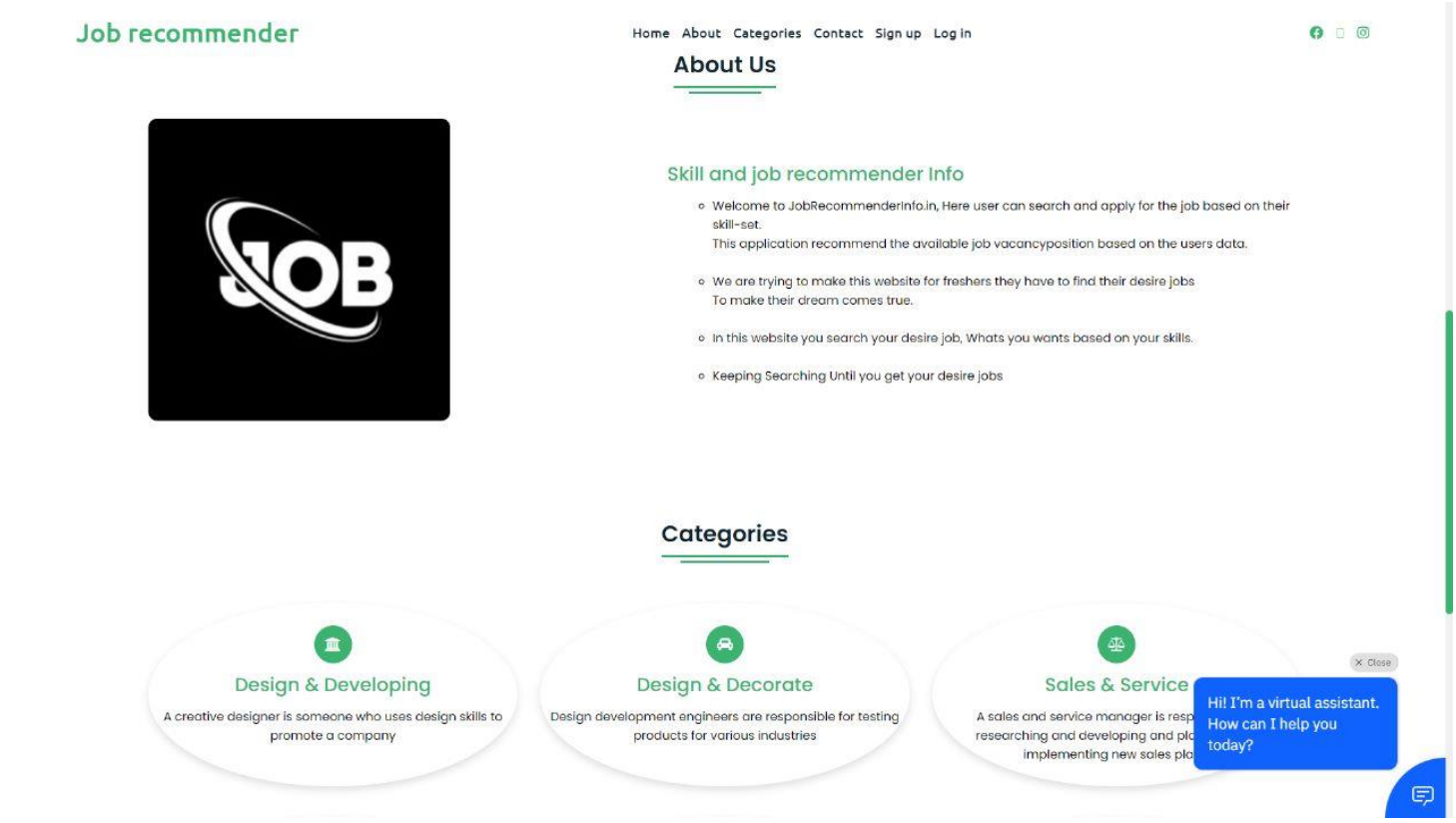

## CHATBOT:



## LOGIN PAGE:




## ABOUT US:



## 13.2 GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-37615-1660313929

## 13.3 PROJECT DEMO LINK:

https://drive.google.com/file/d/19yoUPT5UH4RtrwvAnQ--moE5Xo78nGuJ/view