IBM - PROJECT

Project Documentation Report

Project Name: Skill/Job Recommender Application

ID: PNT2022TMID27264

Team Members:

- **MeenaLochne.S [Team Lead]**
- **Suhail Khan**
- **Akash Vishwakarama**
- **Elizabeth Debbarma**

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-21081-1659771979

1. INTRODUCTION:

Nowadays, searching for a job is frequently done online using job search engines like LinkedIn1, Indeed2, and others. From the beginning of the internet's commercialization in the late 1980s, the question of how this technology could be used in employee recruiting to improve job seeker-vacancy matching was brought up. Many hiring businesses have developed methods for offering the job board in order to support the continuous cycle of the recruiting process from the perspective of the job candidate. Here, a job seeker searches for the positions which he would find relevant to him and apply for it. Due to the large number of job boards, candidates typically choose the one that offers the best services to them, including CV writing, developing a job profile, and suggest new positions to job seekers. Job seekers are looking for new possibilities that fit their talents more actively and persistently. However, it's difficult for companies that target these job seekers to determine the candidate's skill set and offer tailored employment recommendations.

1.1 Project Overview:

We have designed a skill recommender platform that allows both trained and unskilled job seekers to login, search for available positions, or interact with a Chatbot directly to get their ideal position.

To make the job search process simpler for both beginners and experienced candidates. To make the recruitment system more secure and easier and to create a complete web application that can show available jobs based on the user's skill sets. The users and their information are stored in the Database. Based on the user's skill set, an alert is sent when a position is available. By interacting with the chatbot, the user can receive recommendations based on his abilities. To retrieve information about available positions in the market, we can use a job search API that will pull information directly from a website.

1.2 Purpose:

The purpose is to develop a complete online application that can show available positions based on the skillset of the user and aims to help users in finding jobs that match their personnel interests.

An excellent job recommender application not only enables to recommend higher paying job that is most aligned with the skill set of the existing employment, but also makes suggestions on how to acquire a few additional abilities that are necessary to fill the new role.

2. LITERATURE SURVEY:

2.1 Existing problem:

1. Job Recommendation based on Job Profile Clustering and Job Seeker Behaviour
Authors: D. Mahdi*, R. Moulouki, M. Y. El Gourami, M. Azzouazi, L. Moussaid

Project Description:

- Using an automated recommender system, different jobs/skills are Recommended to the user.
- When the application is run, web-scraping techniques are used to gather the data from various job search websites that is processed and examined.
- This model is based on a cluster analysis approach, a self-organized learning technique that aids in grouping job offers based on how similar or different Their qualities are to one another.

Constraints:

- We have discussed certain text clustering techniques and related work in this Study that are linked to automated recommendation before exposing the Principles and guidelines of our suggested model.
- Automated suggestion can help customers identify and select the products based on their needs or based on recommendations from individuals they trust or who have similar likes.
- It uses simple algorithms like Automated Recommendation, which is divided into Content-Based Filtering and Collaborative Filtering.

Possible solutions:

- Based on a job seeker's prior interactions with particular job offers, it will be easier to match a set of job offers to that job seeker.
- The dataset that is to be used is compiled by scraping job-search websites.
- Concentrate on developing and testing the model by utilizing the Word2vec method and the k-means clustering algorithms to collect and reflect the context of job profiles.

2. Job Recommendation based on Job Seeker Skills: An Empirical Study
Authors: Jorge Valverde-Rebaza Ricardo Puma Paul Bustios Nathalie C. Silva
Published Month & Year: June, 2018.

Project Description:

- This model is carried out in the form of a proposed framework based on the Professional skills of job seekers.
- Carried out an evaluation to quantify experimentally the recommendation abilities of two state-of-the-art approaches, considering alternative configurations, within the suggested structure.

Constraints:

- In this part, we provide a brief overview of two techniques that are used in our experiments
- Word2vec general predictive model for learning vector representations of words are called word embedding.
- Term Frequency-Inverse Document Frequency (TF-IDF) is a strategy that has been successful in identifying topics in huge text datasets.
- Additionally, we show two models that are frequently utilized over Word2Vec: Continuous Bag-of-Words (CBOW) and Skip-gram.

Possible Solutions:

The project focuses 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.

3. Job Recommendation System Using Machine Learning and Natural Language Processing

Authors: Dublin Smita Sharma, Nikil Nair, Aditi Patil, Tushar Nayak, and Abhinov Bardan Published Month & Year: March, 2020.

Project Description

- Nowadays, the Recommender system is becoming part of every business.
- The business tries to increase its revenue by raising the users interactive by recommending new items based on user preferences.
- We have witnessed the rise of LinkedIn in the entertainment domain, using their strategies to implement a recommender system into their existing ecosystem.
- There has been a minimal study in the hiring field from the perspective of a job seeker.
- Keywords: Recommender system, Job domain, Content-based filtering Natural language processing, cosine similarity

Constraints:

- This uses the data of the skilled person to get the specific jobs.
- It helps to track the job availability and recommend the job for the client's side approach and consistent- based approach, collaborative approach.

Possible Solutions:

- Based on your skills, it shows the multiple jobs that match your profile.
- The recruiters hire job seekers based on their LinkedIn profile that matches the job requirement.

4. A Research of Job Recommendation System Based on Collaborative Filtering

Authors: Yingya Zhang; Cheng Yang; Zhixiang Niu

Published Year & Month: May, 2014

Project Description:

- Dealing with the enormous amount of recruiting information on the Internet a job seeker always spends hours to find useful ones.
- In order to reduce this time-consuming work, it is required to design and

Implement a recommendation system for online job-hunting.

Constraints:

This project deliberately uses two approaches:

- Constraint user-based filtering algorithm
- Item-based collaborative filtering algorithm

Possible Solution:

It helps to keep the background information and student’s resume by which the user can apply for the candidate jobs using the recommendation system.

2.2 References:

1. https://smartinternz.com/saas-guided-project/1/skill-job-recommender

2. https://github.com/topics/job-recommendation

3. D. Mhamdi, R. Moulouki, M.Y. El Ghoumari, M. Azzouazi
   Job Recommendation System based on Text Analysis
   Jour of Adv Research in Dynamical & Control Systems, 12 (04) (2020)

2.3 Problem Statement Definition:

A job seeker always spends hours looking through the massive amount of recruiting information on the Internet to find ones that are helpful.

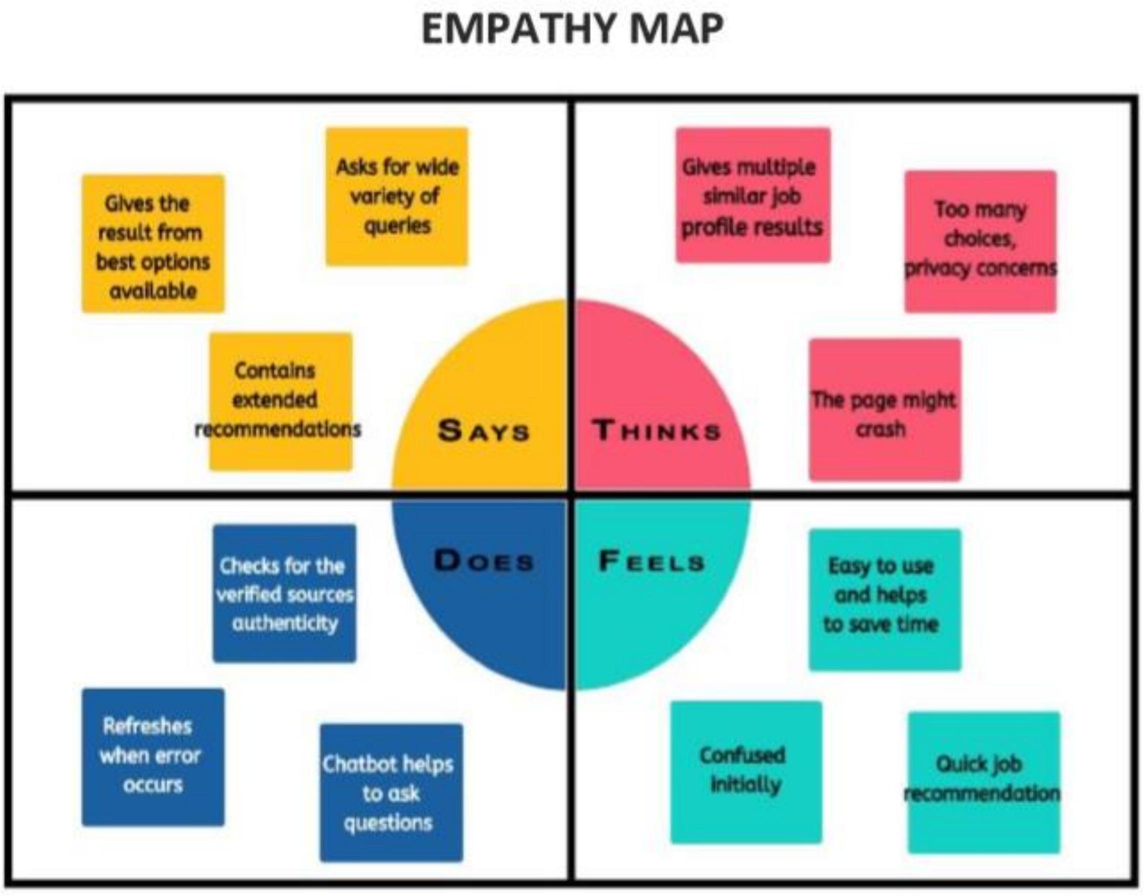
People who don't have any industry experience often don't know exactly what they need to learn in order to get a career that's right for them. We address the issue of suggesting suitable occupations to those looking for a new career. To reduce this laborious work, we design and implement a recommendation system for online job hunting.

3. IDEATION & PROPOSED SOLUTION:

3.1 Empathy Map Canvas:

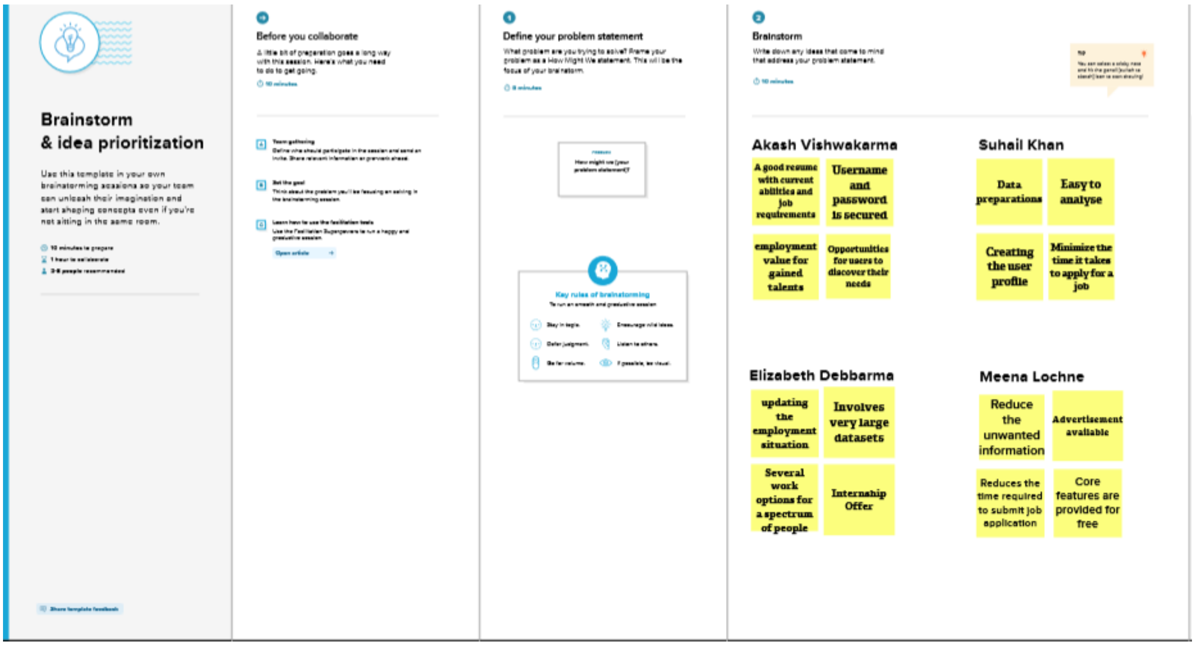
An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

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.


EMPATHY MAP
Gives the result from best options available
Asks for wide variety of queries
Contains extended recommendations
SAYS
Gives multiple similar job profile results
Too many choices, privacy concerns
The page might crash
THINKS
DOES
Checks for the verified sources authenticity
Refreshes when error occurs
Chatbot helps to ask questions
FEELS
Easy to use and helps to save time
Confused initially
Quick job recommendation

3.2 Ideation & Brainstorming:

Ideation is often closely related to the practice of brainstorming, a specific technique that is utilized to generate new ideas. A principal difference between ideation and brainstorming is that ideation is commonly more thought of as being an individual pursuit, while brainstorming is almost always a group activity.



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

- 10 minutes to prepare
- 1 hour to collaborate
- 3-8 people recommended

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

- Team gathering: Define who should participate in the session and send an invite. Share relevant information or prework ahead.
- Set the goal: Think about the problem you'll be focusing on solving in the brainstorming session.
- Learn how to use the facilitation tools: Use the Facilitation Superpowers to run a happy and productive session.

Open article →

Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

How might we [your problem statement]?

Key rules of brainstorming

To run an smooth and productive session

- Stay in topic.
- Encourage wild ideas.
- Defer judgment.
- Listen to others.
- Go for volume.
- If possible, be visual.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

Akash Vishwakarma

- A good resume with current abilities and job requirements
- Username and password is secured
- employment value for gained talents
- Opportunities for users to discover their needs

Suhail Khan

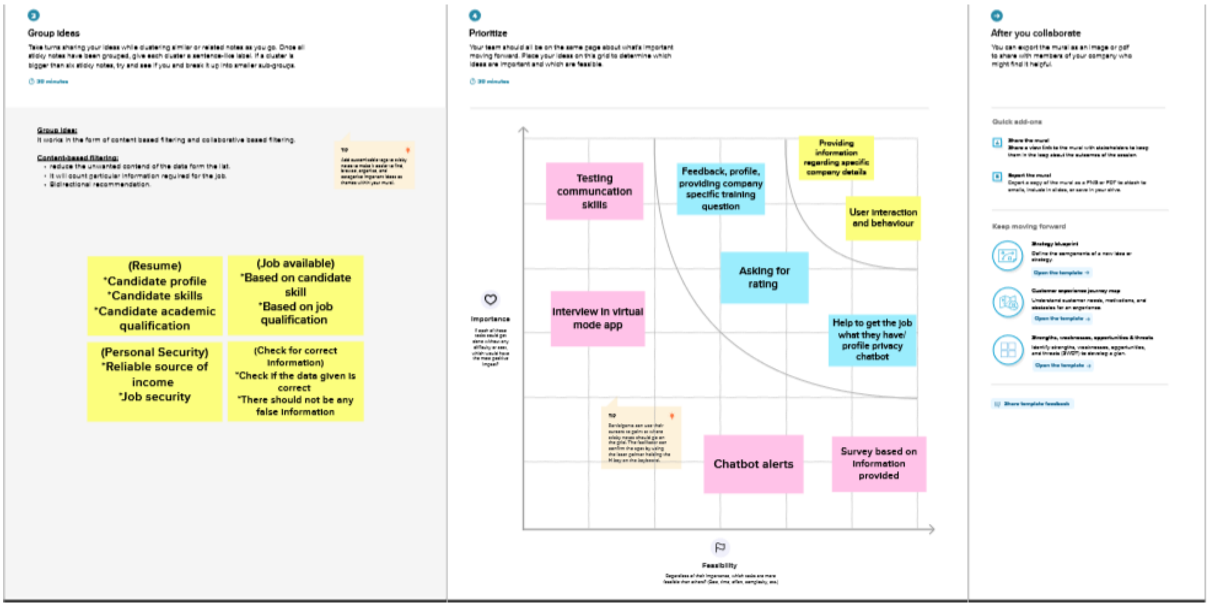
- Data preparations
- Easy to analyse
- Creating the user profile
- Minimize the time it takes to apply for a job

Elizabeth Debbarma

- updating the employment situation
- Involves very large datasets
- Several work options for a spectrum of people
- Internship Offer

Meena Lochne

- Reduce the unwanted information
- Advertisement available
- Reduces the time required to submit job application
- Core features are provided for free



Group Ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

Group Ideas:
It works in the form of content based filtering and collaborative based filtering.

Content-based filtering:
- reduce the unwanted contend of the data form the list.
- It will count particular information required for the job.
- Bidirectional recommendation.

(Resume)
*Candidate profile
*Candidate skills
*Candidate academic qualification

(Job available)
*Based on candidate skill
*Based on job qualification

(Personal Security)
*Reliable source of income
*Job security

(Check for correct information)
*Check if the data given is correct
*There should not be any false information

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Importance
Feasibility
Testing communcation skills
Feedback, profile, providing company specific training question
Providing information regarding specific company details
User interaction and behaviour
Asking for rating
Interview in virtual mode app
Help to get the job what they have/ profile privacy chatbot
Chatbot alerts
Survey based on information provided

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

- Share the mural: Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
- Export the mural: Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

- Strategy blueprint: Define the components of a new idea or strategy. Open the template →
- Customer experience journey map: Understand customer needs, motivations, and obstacles for an experience. Open the template →
- Strengths, weaknesses, opportunities & threats: Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan. Open the template →

Share template feedback

3.3 Proposed Solution:

Project team shall fill the following information in proposed solution template.

1. Problem Statement (Problem to be solved):

- ✓ To recommend jobs based on skills of the users.
- ✓ This application gives the best results from the best options that are available to them.
- ✓ The application asks the user for a wide variety of questions to set up the results on those grounds. It also shows up intended recommendations that could be helpful in some sense to the user.

2. Idea / Solution description

- ✓ By matching the skills of the user with the already available profiles on the application and if a unique skill set appears then based on the jobs with the maximum stretch of the profile is recommended to the user.

3. Novelty / Uniqueness

- ✓ Skill job recommendation is a system that will keep track of the profiles based on the skills that they put up.
- ✓ By adding verified sources of the skill certificates, authenticity of the user is also prioritized. It can also link with other profiles on the system. The users can contact the orgs based on the recommendation by the system.
- ✓ System contains a chatbot to help resolve queries based on the problems that the user may cover across while going through the application.

4. Social Impact / Customer Satisfaction

- ✓ Verified sources of jobs
- ✓ Easy to use ,helps save time as well
- ✓ Availability of wide sources

5. Business Model (Revenue Model)

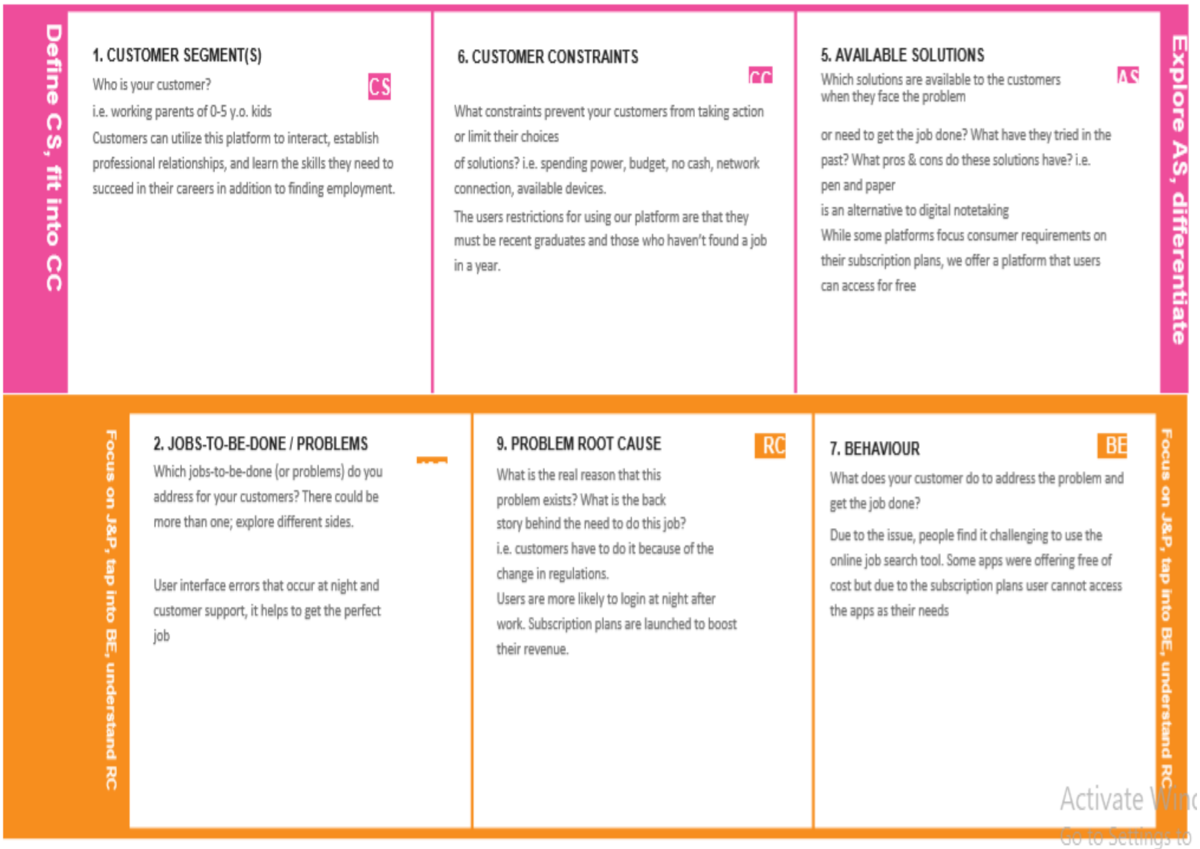
- ✓ Asks for a subscription to add extra skills that gives edge from other users.
- ✓ We can add for an amount to search for recommendations after a certain number of search attempts.
- ✓ When the subscription gets over, the difference would be shown to the user how he/she got special treatment in comparison to others.

6. Scalability of the Solution

- ✓ It could be increased by increasing the collaboratively with external apps or directly associating with new job sources

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.



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

i.e. working parents of 0-5 y.o. kids

Customers can utilize this platform to interact, establish professional relationships, and learn the skills they need to succeed in their careers in addition to finding employment.

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.

The users restrictions for using our platform are that they must be recent graduates and those who haven't found a job in a year.

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

While some platforms focus consumer requirements on their subscription plans, we offer a platform that users can access for free

Explore AS, differentiate

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

2. JOBS-TO-BE-DONE / PROBLEMS

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

User interface errors that occur at night and customer support, it helps to get the perfect job

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.

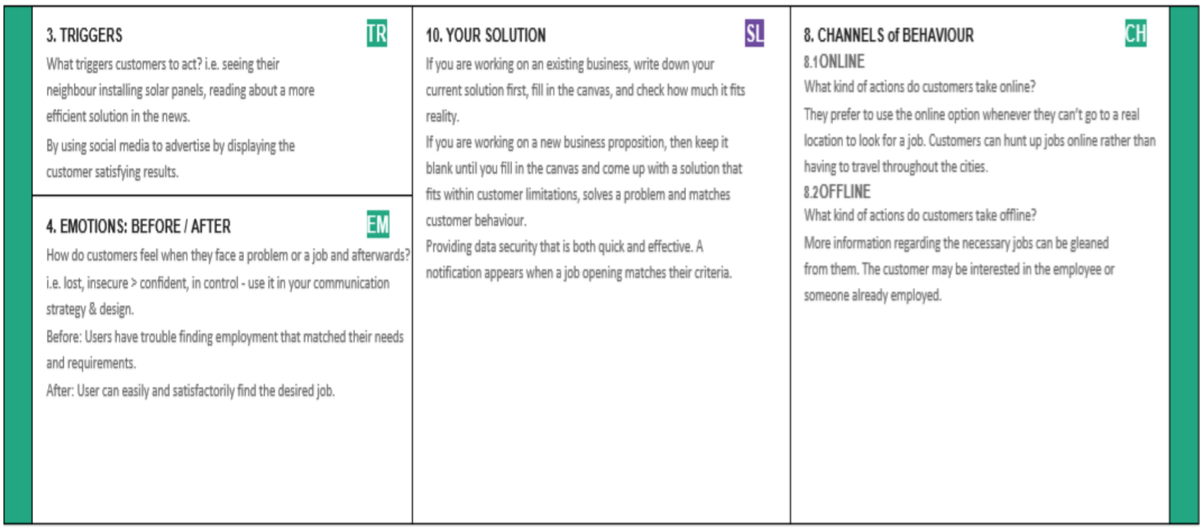
Users are more likely to login at night after work. Subscription plans are launched to boost their revenue.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Due to the issue, people find it challenging to use the online job search tool. Some apps were offering free of cost but due to the subscription plans user cannot access the apps as their needs

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



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.

By using social media to advertise by displaying the customer satisfying results.

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: Users have trouble finding employment that matched their needs and requirements.

After: User can easily and satisfactorily find the desired 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.

Providing data security that is both quick and effective. A notification appears when a job opening matches their criteria.

8. CHANNELS of BEHAVIOUR CH

8.1ONLINE

What kind of actions do customers take online?

They prefer to use the online option whenever they can't go to a real location to look for a job. Customers can hunt up jobs online rather than having to travel throughout the cities.

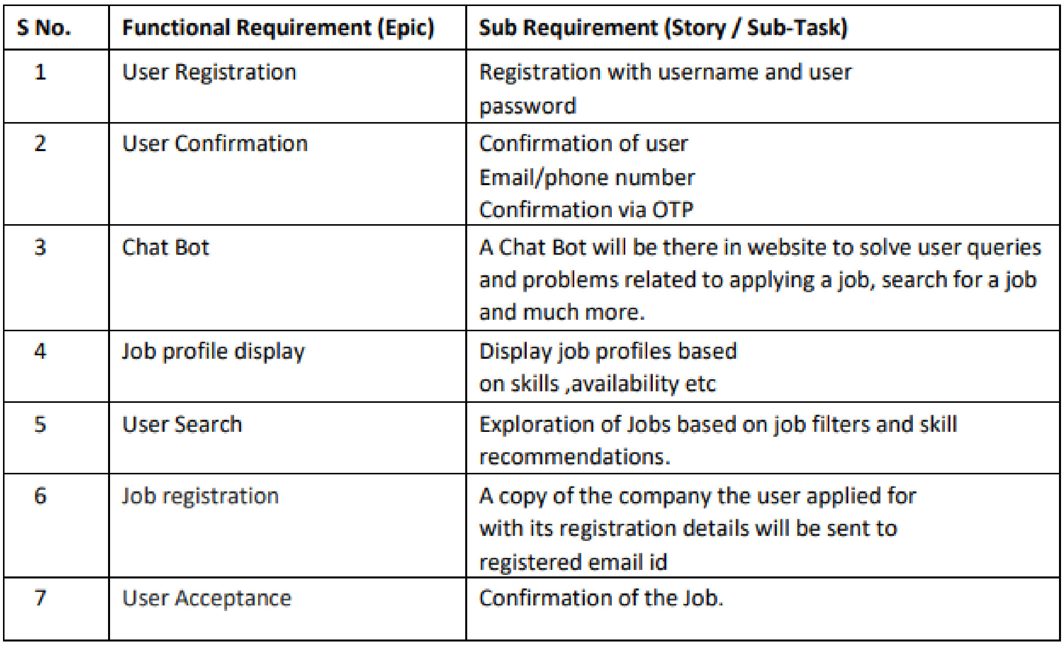
8.2OFFLINE

What kind of actions do customers take offline?

More information regarding the necessary jobs can be gleaned from them. The customer may be interested in the employee or someone already employed.

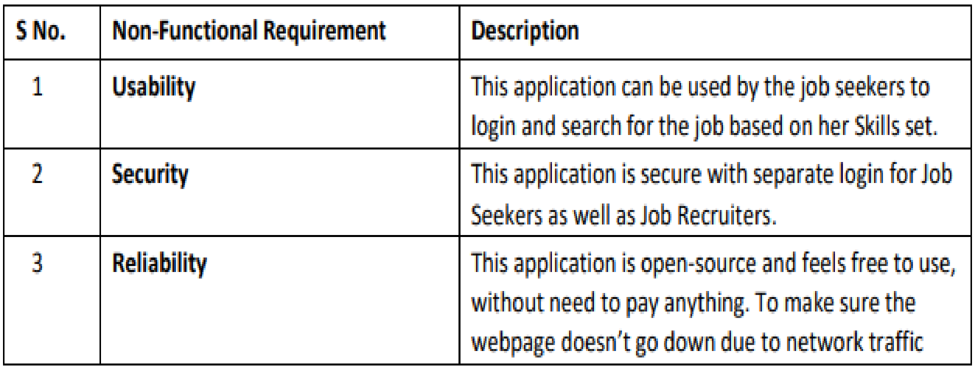
4. REQUIREMENT ANALYSIS:

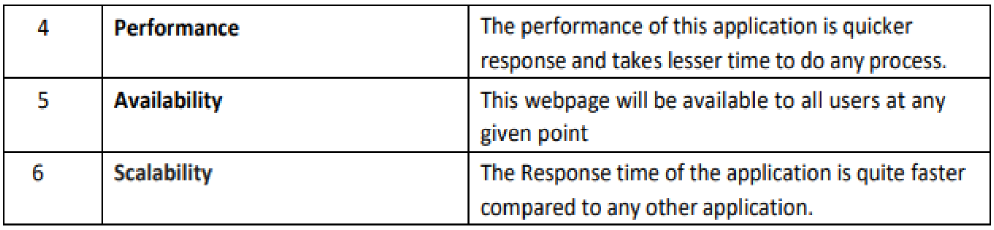
4.1Functional requirement:



S No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
1	User Registration	Registration with username and user password
2	User Confirmation	Confirmation of user Email/phone number Confirmation via OTP
3	Chat Bot	A Chat Bot will be there in website to solve user queries and problems related to applying a job, search for a job and much more.
4	Job profile display	Display job profiles based on skills ,availability etc
5	User Search	Exploration of Jobs based on job filters and skill recommendations.
6	Job registration	A copy of the company the user applied for with its registration details will be sent to registered email id
7	User Acceptance	Confirmation of the Job.

4.2Non-Functional requirements:





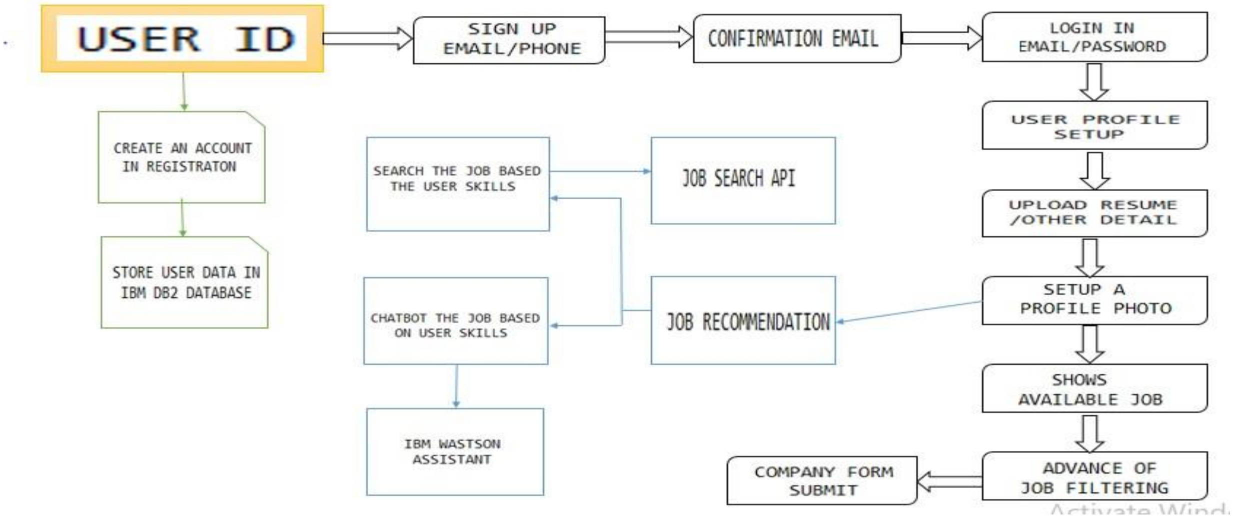
S No.	Non-Functional Requirement	Description
1	**Usability**	This application can be used by the job seekers to login and search for the job based on her Skills set.
2	**Security**	This application is secure with separate login for Job Seekers as well as Job Recruiters.
3	**Reliability**	This application is open-source and feels free to use, without need to pay anything. To make sure the webpage doesn't go down due to network traffic
4	**Performance**	The performance of this application is quicker response and takes lesser time to do any process.
5	**Availability**	This webpage will be available to all users at any given point
6	**Scalability**	The Response time of the application is quite faster compared to any other application.

5. PROJECT DESIGN:

5.1 Data Flow Diagram:

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.

Example:



User Stories

Use the below template to list all the user stories for the product

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 phone number.	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the		Medium	Sprint-1

			application through Gmail.			
	Login	USN-5	As a user, I can log into the application by entering email & password.		High	Sprint-1
	Dashboard	USN-6	As a user, I will enter the detail about my profile in the dashboard then sign in.	I can access my account / dashboard	Low	Sprint-2
	User Detail	USN-7	As a user, I will upload all the details and resume, certificates, skills developed, working experience.	I can view the next page to enter the detail	Low	Sprint-2
		USN-8	As a user, once I completed the first steps then can setup a profile photos and basic details by signing in	I can prepare multiple task in the application	Mediu m	Sprint-3

5.2 Solution and Technical Architecture:

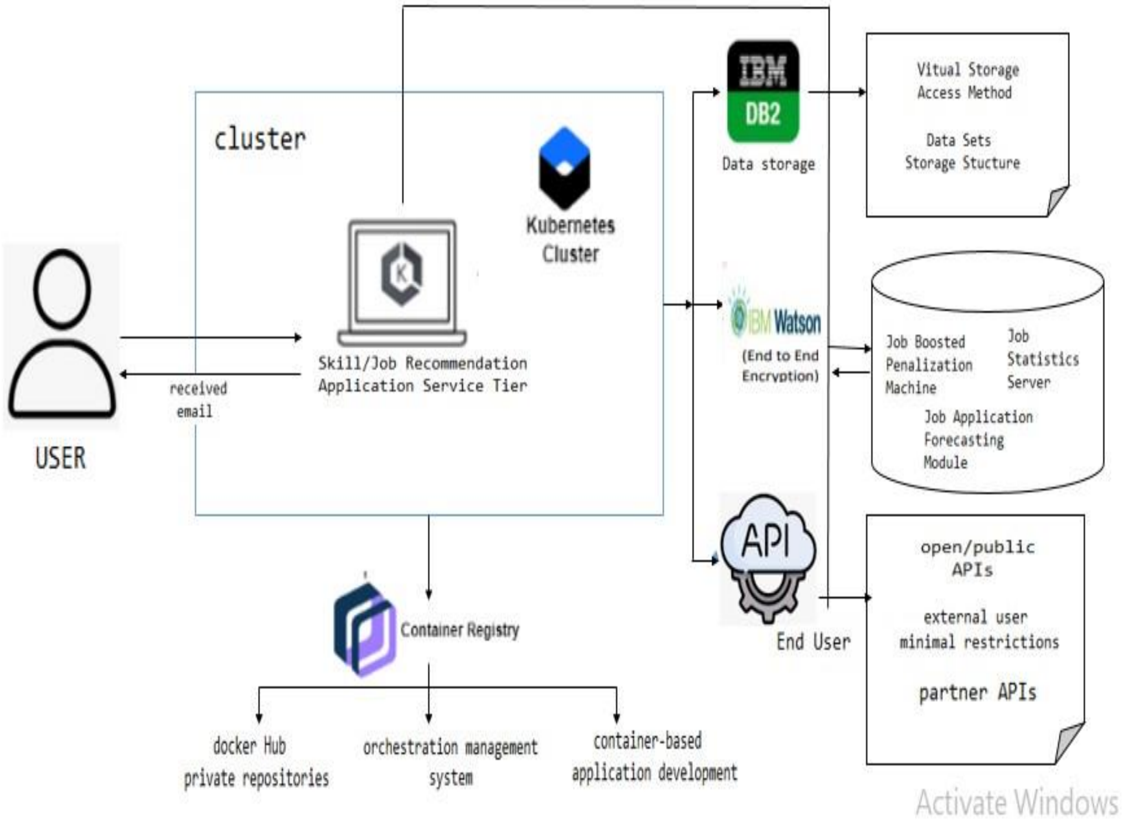
Solution Architecture:

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

Its goals are to:

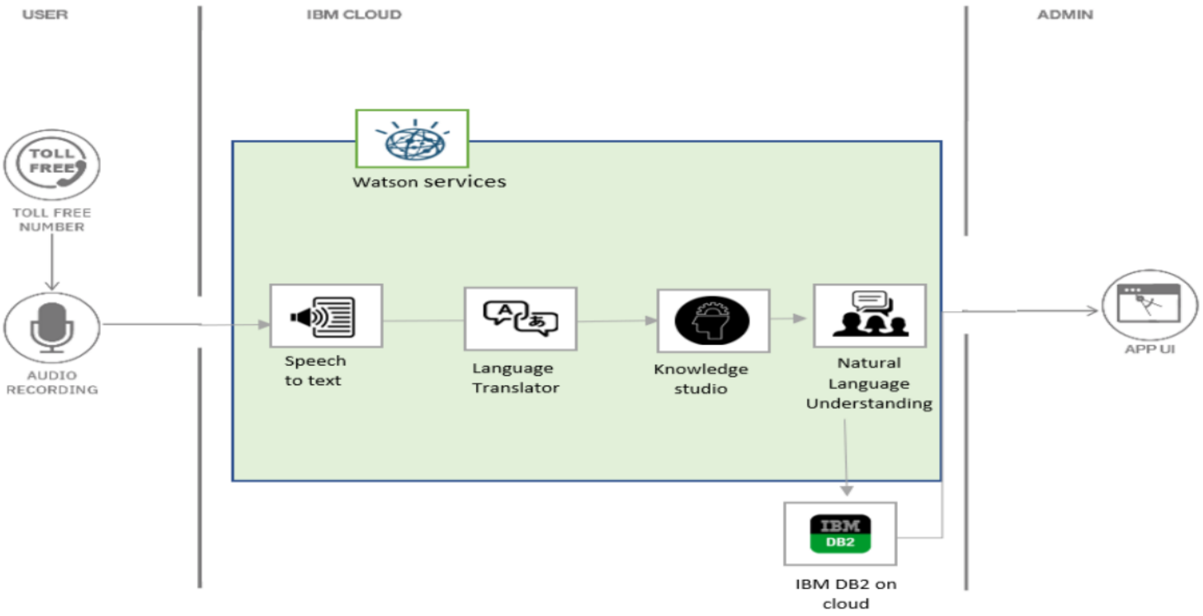
- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behaviour, and other aspects of the software to project stakeholders.
- Define features, development phases, and solution requirements.
- Provide specifications according to which the solution is defined, managed, and delivered.

Solution Architecture Diagram:


cluster
Kubernetes Cluster
USER
received email
Skill/Job Recommendation Application Service Tier
Container Registry
docker Hub private repositories
orchestration management system
container-based application development
IBM DB2
Data storage
Vitual Storage Access Method
Data Sets Storage Stucture
IBM Watson
(End to End Encryption)
Job Boosted Penalization Machine
Job Statistics Server
Job Application Forecasting Module
API
End User
open/public APIs
external user minimal restrictions
partner APIs
Activate Windows

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



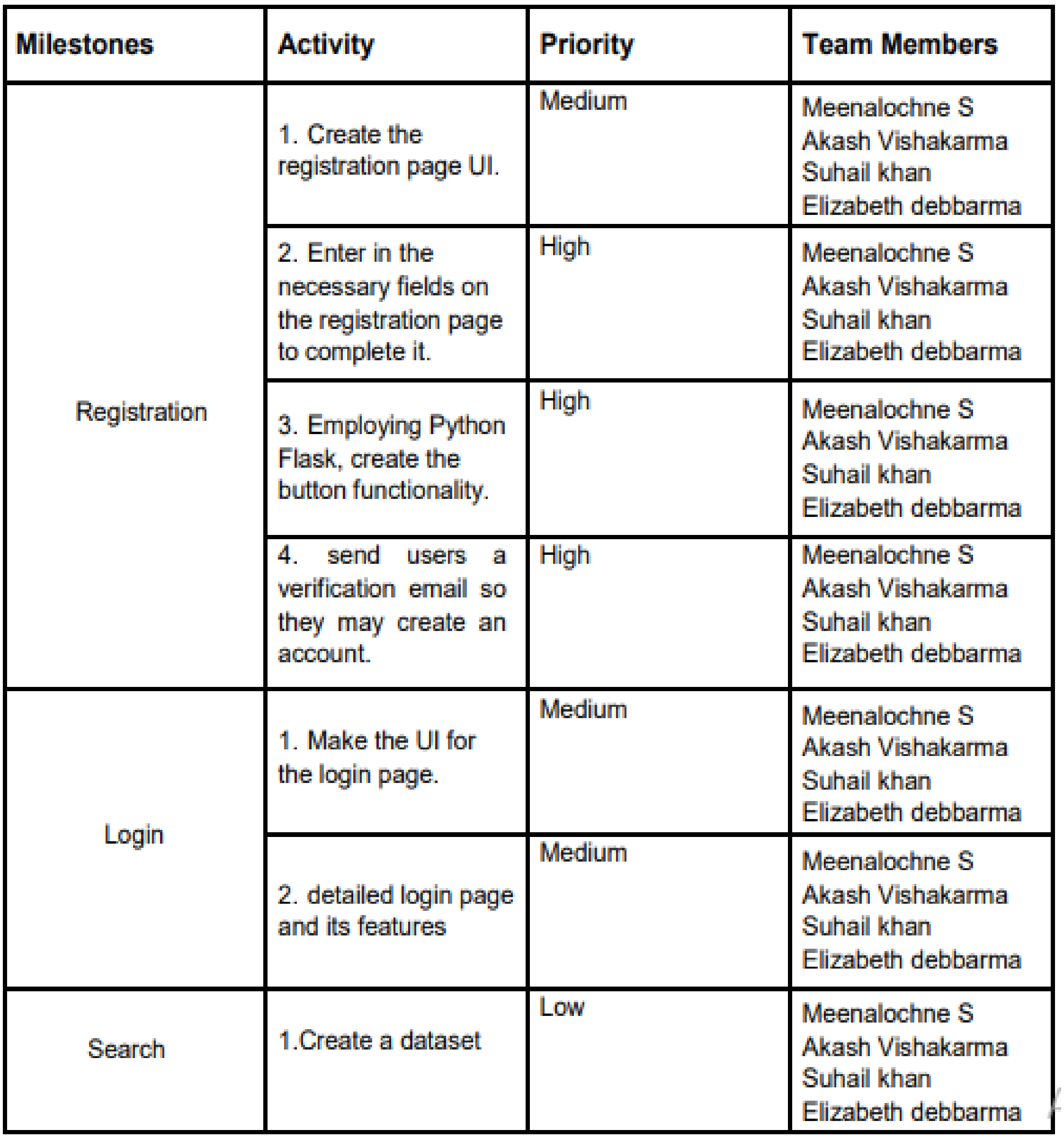
S. No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Application Logic-1	Logic for a process in the application	Java / Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL, NoSQL, etc.
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloud and etc.
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local File system
8.	External API-1	Purpose of External API used in the application	IBM Weather API, etc.
9.	External API-2	Purpose of External API used in the application	Aadhar API, etc.
10.	Machine Learning Model	Purpose of Machine Learning Model	Object Recognition Model, etc.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Open source framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used

5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

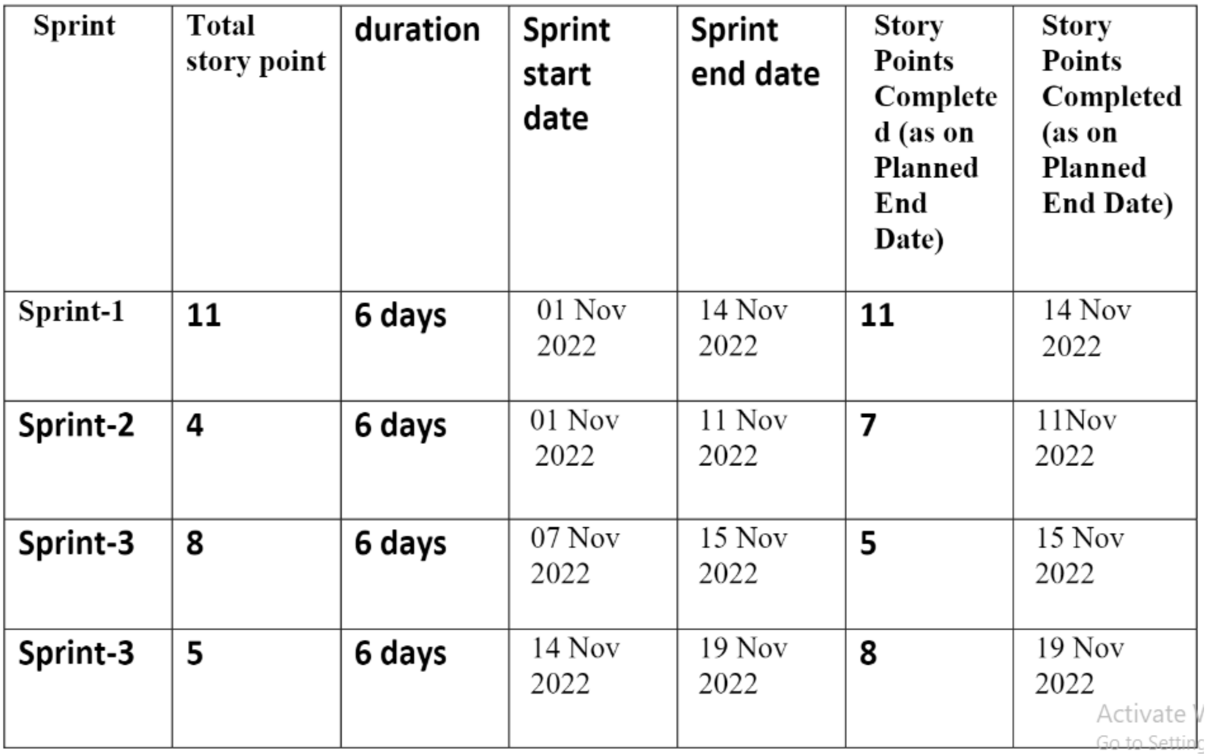
6. PROJECT PLANNING & SCHEDULING:

6.1 Sprint Planning & Estimation:



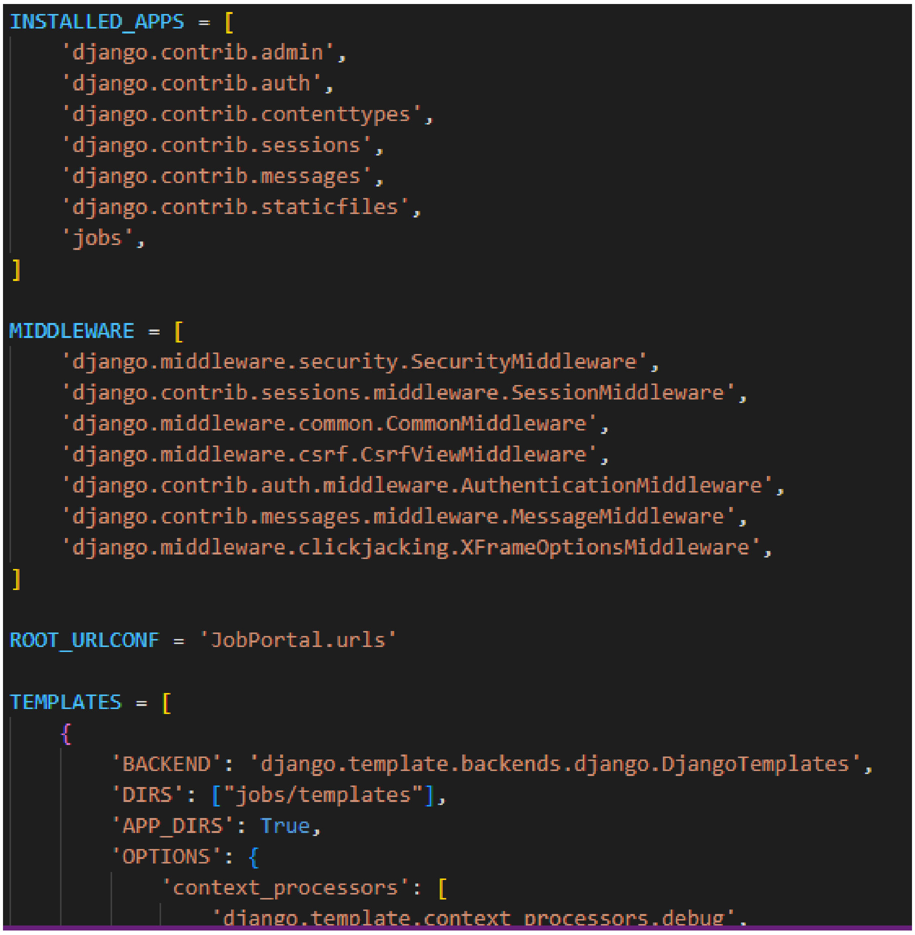
Milestones	Activity	Priority	Team Members
Registration	1. Create the registration page UI.	Medium	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma
	2. Enter in the necessary fields on the registration page to complete it.	High	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma
	3. Employing Python Flask, create the button functionality.	High	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma
	4. send users a verification email so they may create an account.	High	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma
Login	1. Make the UI for the login page.	Medium	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma
	2. detailed login page and its features	Medium	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma
Search	1.Create a dataset	Low	Meenalochne S Akash Vishakarma Suhail khan Elizabeth debbarma

6.1Sprint Delivery Schedule :

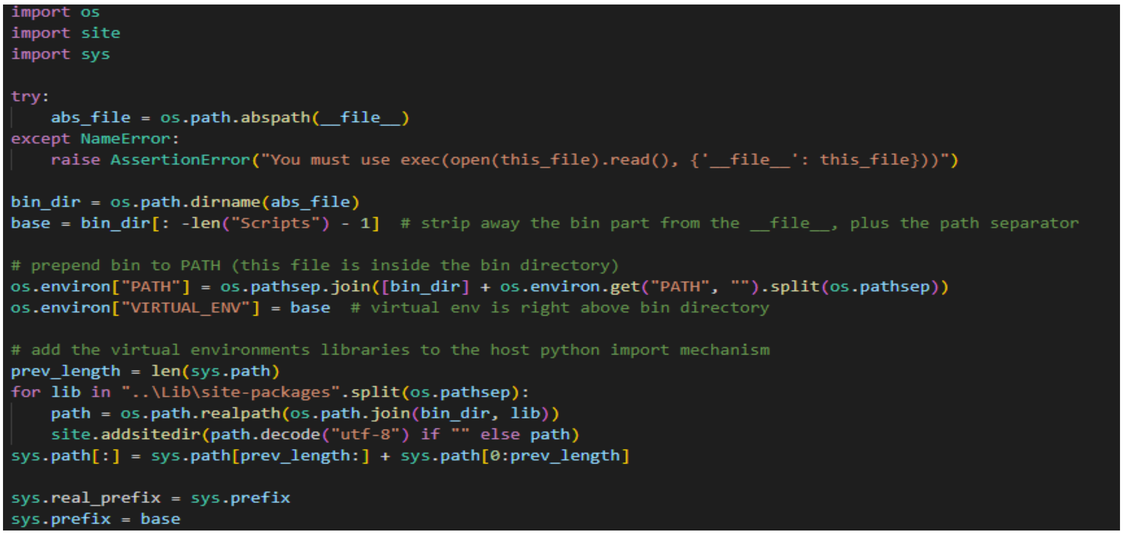


Sprint	Total story point	duration	Sprint start date	Sprint end date	Story Points Completed (as on Planned End Date)	Story Points Completed (as on Planned End Date)
Sprint-1	11	6 days	01 Nov 2022	14 Nov 2022	11	14 Nov 2022
Sprint-2	4	6 days	01 Nov 2022	11 Nov 2022	7	11Nov 2022
Sprint-3	8	6 days	07 Nov 2022	15 Nov 2022	5	15 Nov 2022
Sprint-3	5	6 days	14 Nov 2022	19 Nov 2022	8	19 Nov 2022

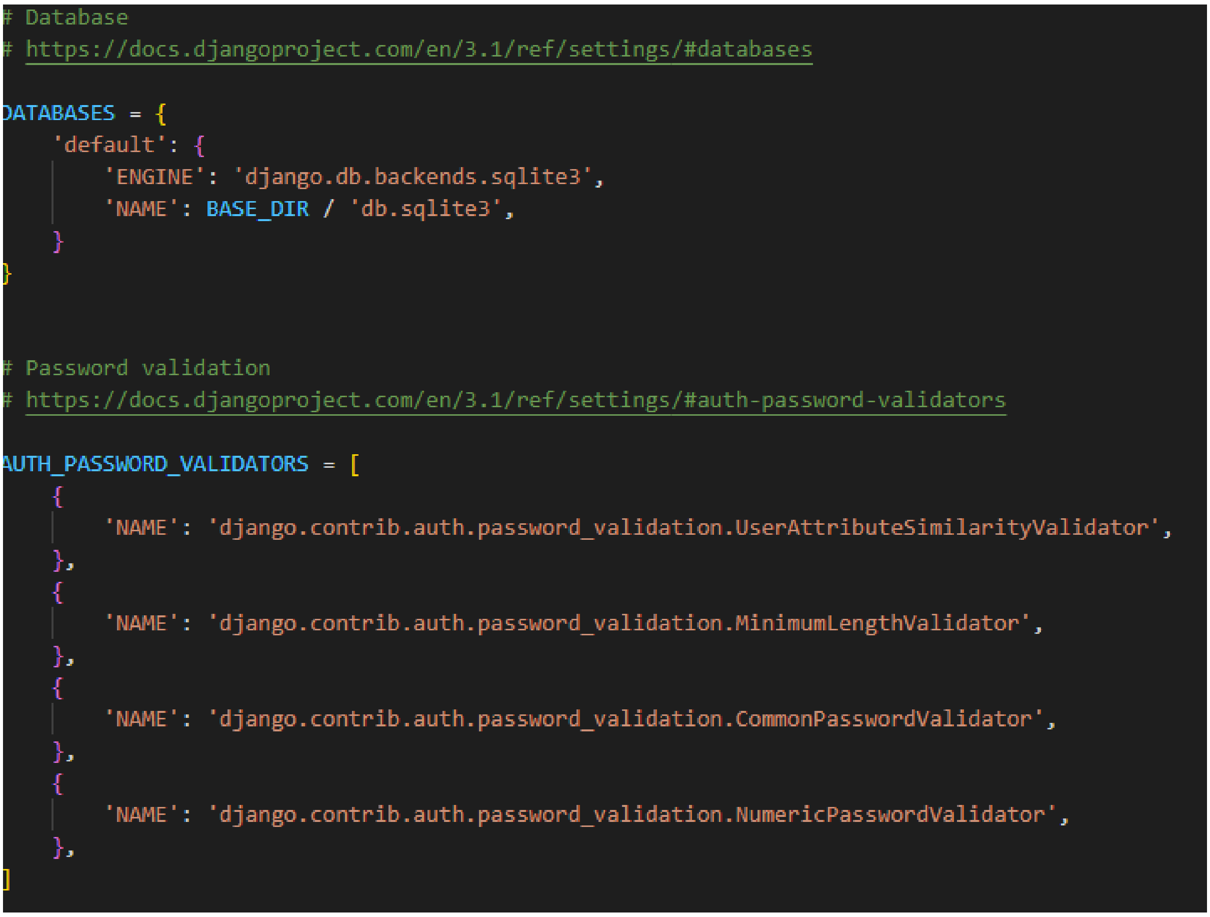
7. CODING & SOLUTIONING:



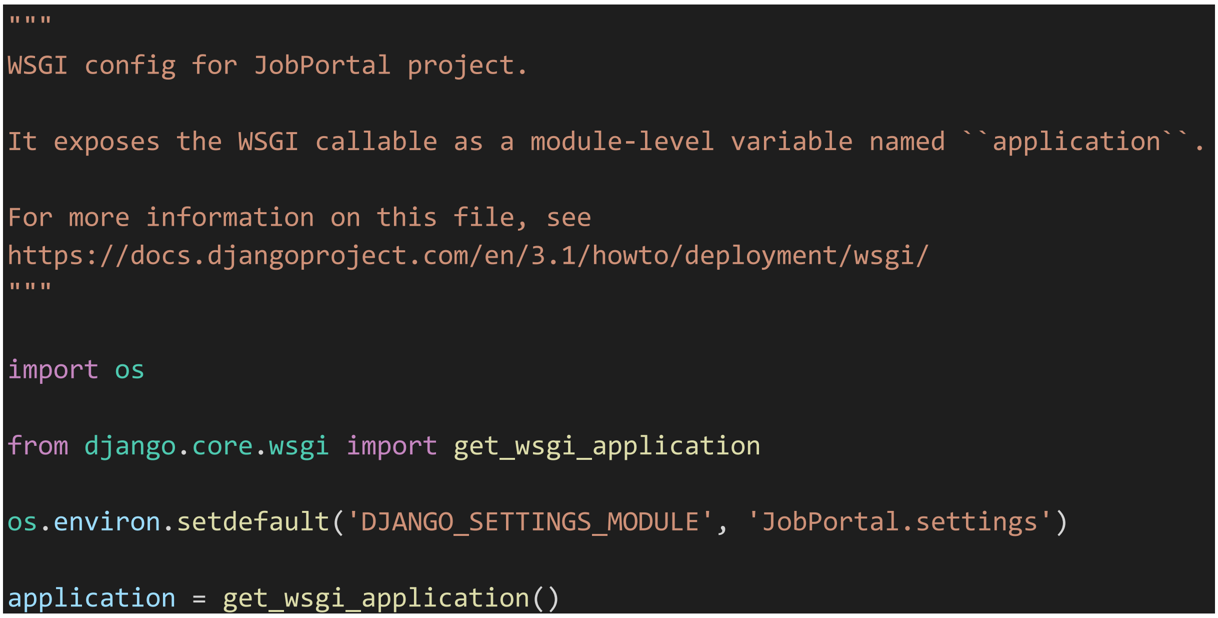
```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'jobs',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'JobPortal.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ["jobs/templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
```



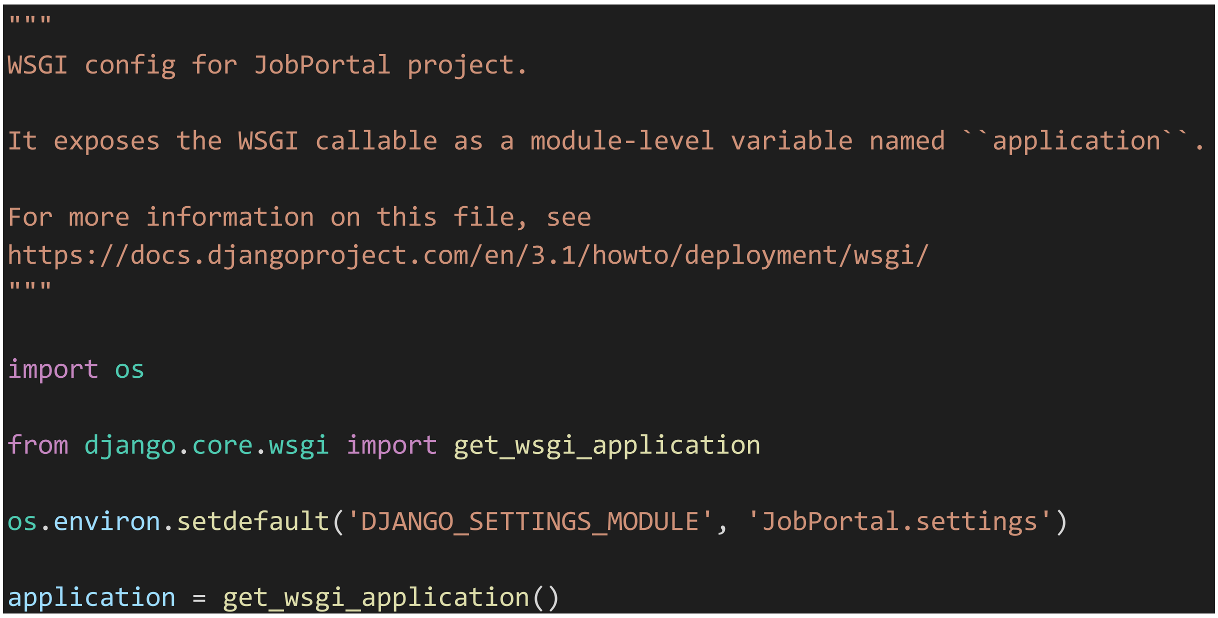
```
import os
import site
import sys

try:
    abs_file = os.path.abspath(__file__)
except NameError:
    raise AssertionError("You must use exec(open(this_file).read(), {'__file__': this_file}))")

bin_dir = os.path.dirname(abs_file)
base = bin_dir[: -len("Scripts") - 1]  # strip away the bin part from the __file__, plus the path separator

# prepend bin to PATH (this file is inside the bin directory)
os.environ["PATH"] = os.pathsep.join([bin_dir] + os.environ.get("PATH", "").split(os.pathsep))
os.environ["VIRTUAL_ENV"] = base  # virtual env is right above bin directory

# add the virtual environments libraries to the host python import mechanism
prev_length = len(sys.path)
for lib in "..\Lib\site-packages".split(os.pathsep):
    path = os.path.realpath(os.path.join(bin_dir, lib))
    site.addsitedir(path.decode("utf-8") if "" else path)
sys.path[:] = sys.path[prev_length:] + sys.path[0:prev_length]

sys.real_prefix = sys.prefix
sys.prefix = base
```



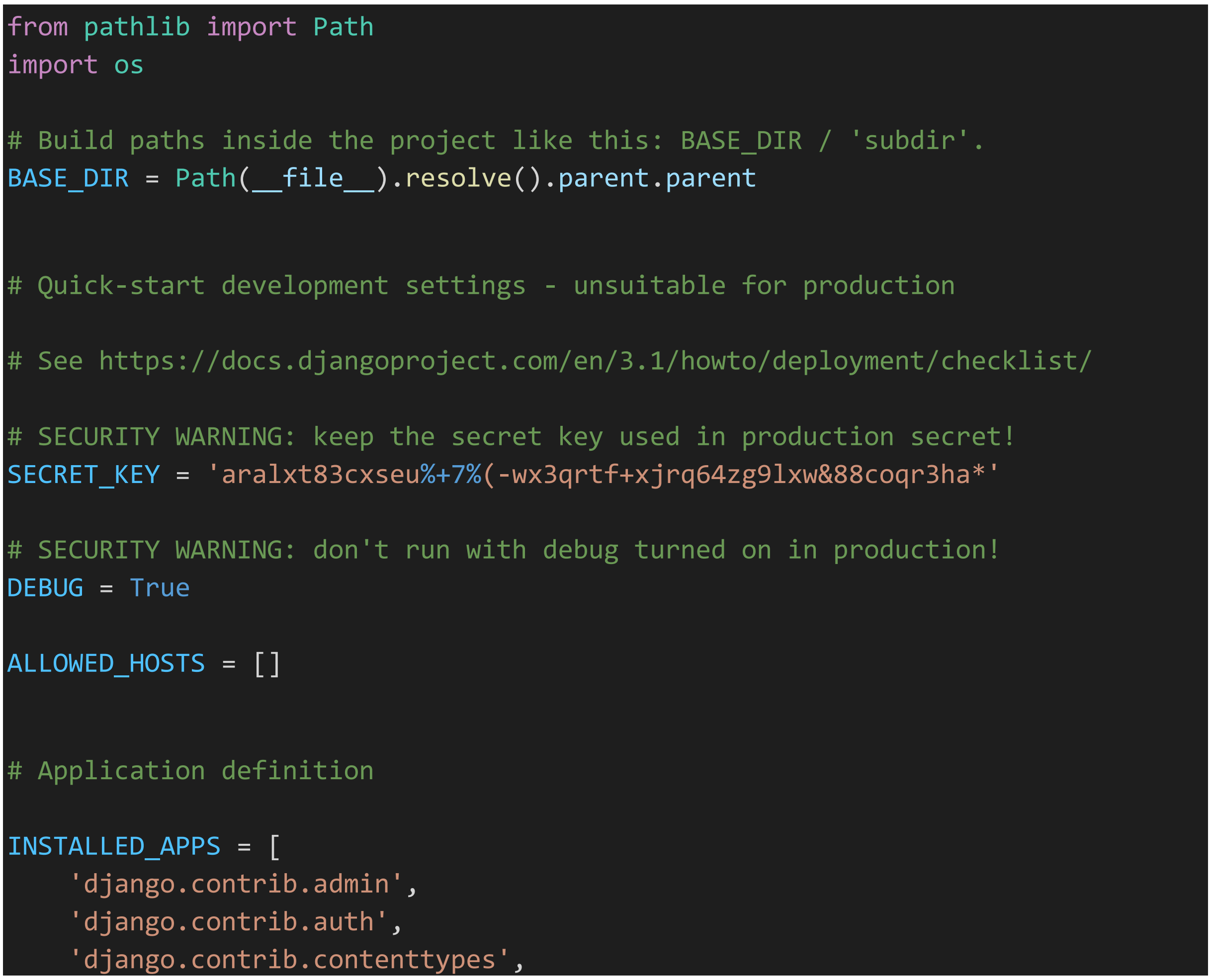
```
# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}


# Password validation
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```



```
"""
WSGI config for JobPortal project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'JobPortal.settings')

application = get_wsgi_application()
```



```
"""
WSGI config for JobPortal project.

It exposes the WSGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/wsgi/
"""

import os

from django.core.wsgi import get_wsgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'JobPortal.settings')

application = get_wsgi_application()
```



```
from pathlib import Path
import os

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production

# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'aralxt83cxseu%+7%(-wx3qrtf+xjrq64zg9lxw&88coqr3ha*'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'jobs',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'JobPortal.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': ["jobs/templates"],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },

    },
]

WSGI_APPLICATION = 'JobPortal.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.1/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': BASE_DIR / 'db.sqlite3',
    }
}
```

```
# Password validation
# https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]


# Internationalization
# https://docs.djangoproject.com/en/3.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True




USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.1/howto/static-files/

STATIC_URL = '/static/'

STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "jobs/static")
]
```

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'jobs/media')
MEDIA_URL = '/media/'
```

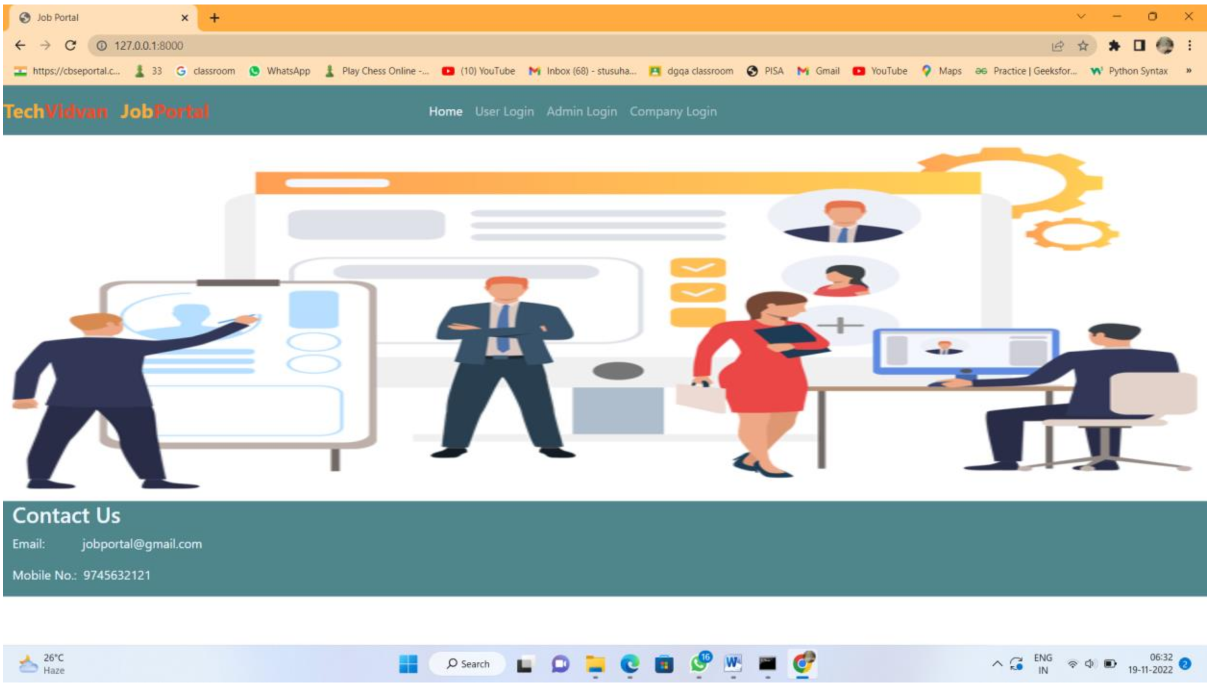
```
"""
ASGI config for JobPortal project.

It exposes the ASGI callable as a module-level variable named ``application``.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/howto/deployment/asgi/
"""

import os

from django.core.asgi import get_asgi_application

os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'JobPortal.settings')

application = get_asgi_application()
```

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys


def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'JobPortal.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)
```

```
if __name__ == '__main__':
    main()
```

-

8.TESTING

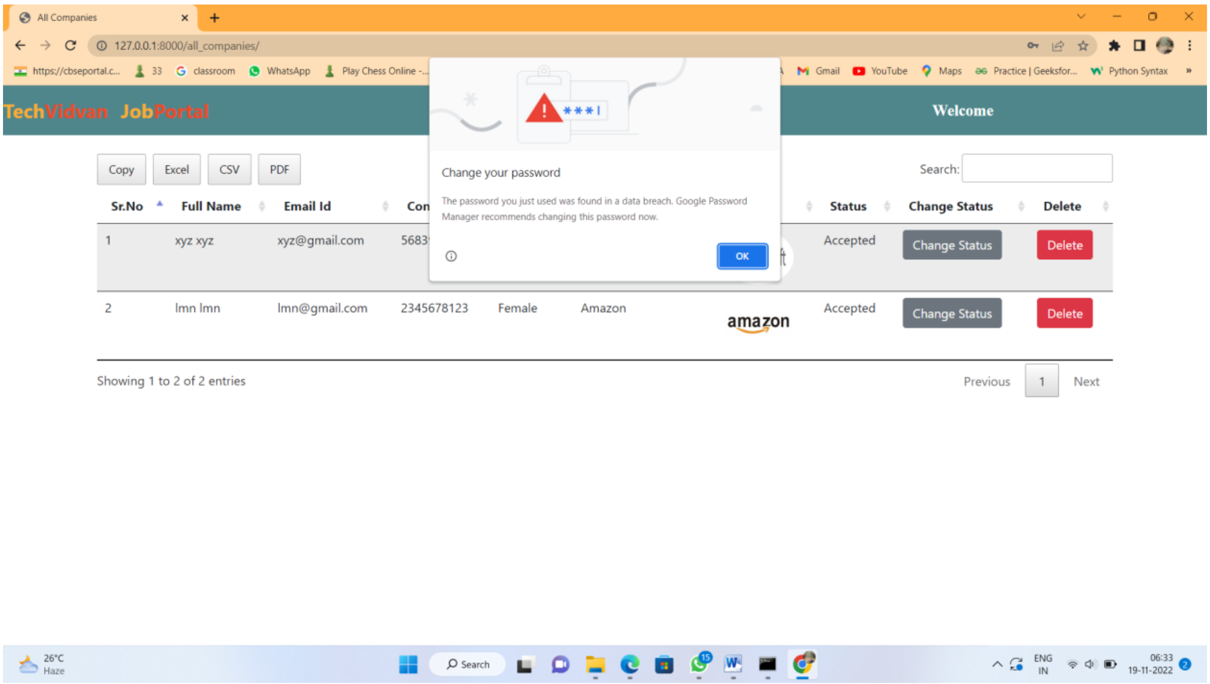

Job Portal
127.0.0.1:8000
https://cbseportal.c... 33 classroom WhatsApp Play Chess Online -... (10) YouTube Inbox (68) - stusuha... dgqa classroom PISA Gmail YouTube Maps Practice | Geeksfor... Python Syntax
Home
User Login
Admin Login
Company Login
26°C
Haze
Search
ENG
IN
06:32
19-11-2022

TechVidvan JobPortal

Contact Us

Email: jobportal@gmail.com

Mobile No.: 9745632121


All Companies
127.0.0.1:8000/all_companies/
https://cbseportal.c... 33 classroom WhatsApp Play Chess Online -... Gmail YouTube Maps Practice | Geeksfor... Python Syntax
Copy
Excel
CSV
PDF
Search:
OK
Previous
1
Next
26°C
Haze
Search
ENG
IN
06:33
19-11-2022

TechVidvan JobPortal

Welcome

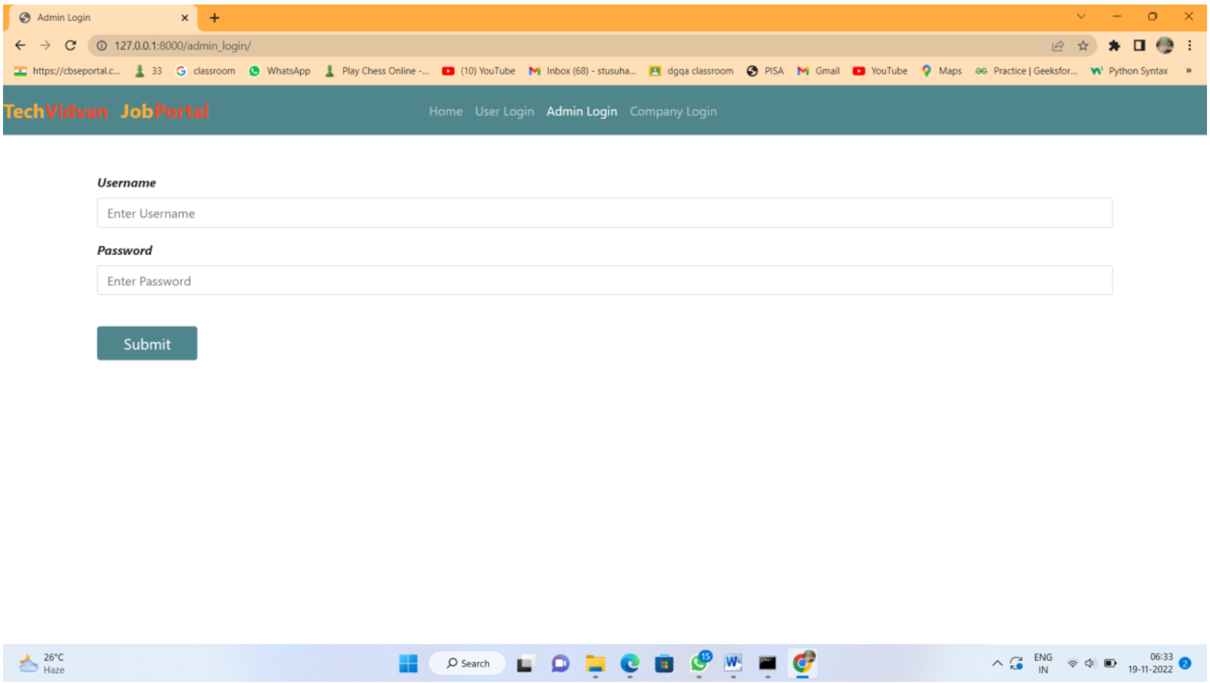
Change your password

The password you just used was found in a data breach. Google Password Manager recommends changing this password now.

Sr.No	Full Name	Email Id	Con				Status	Change Status	Delete
1	xyz xyz	xyz@gmail.com	5683				Accepted	Change Status	Delete
2	lmn lmn	lmn@gmail.com	2345678123	Female	Amazon	amazon	Accepted	Change Status	Delete

Showing 1 to 2 of 2 entries

9.RESULTS

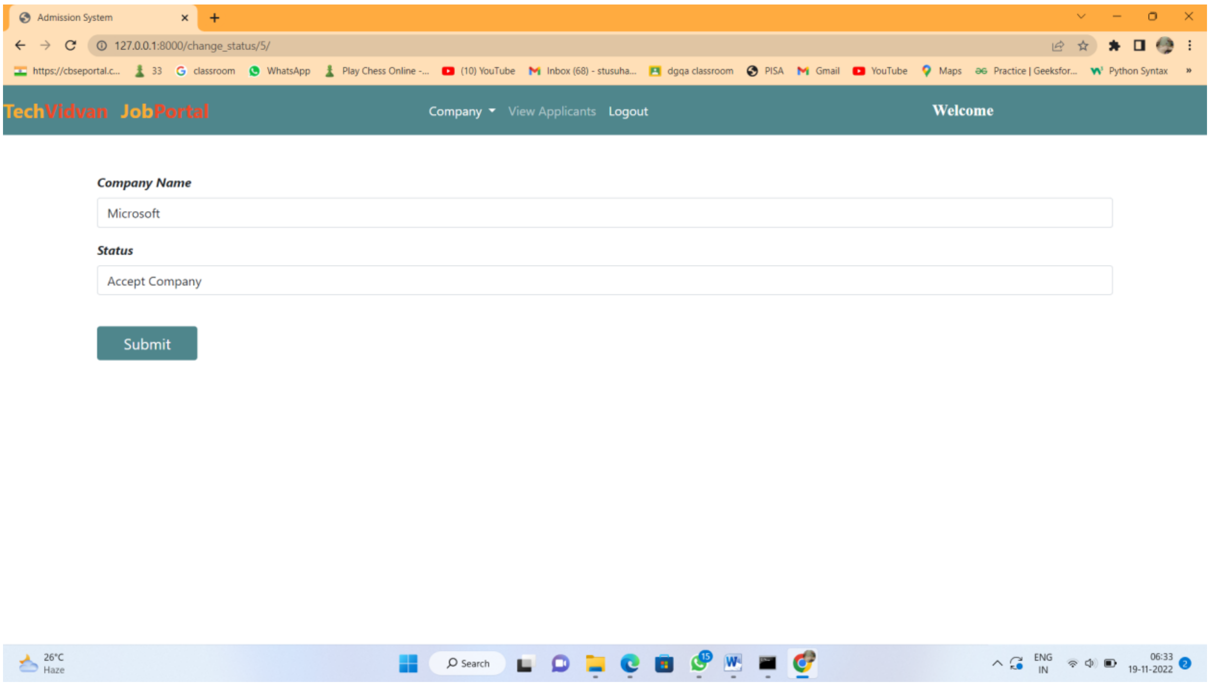


Username

Enter Username

Password

Enter Password

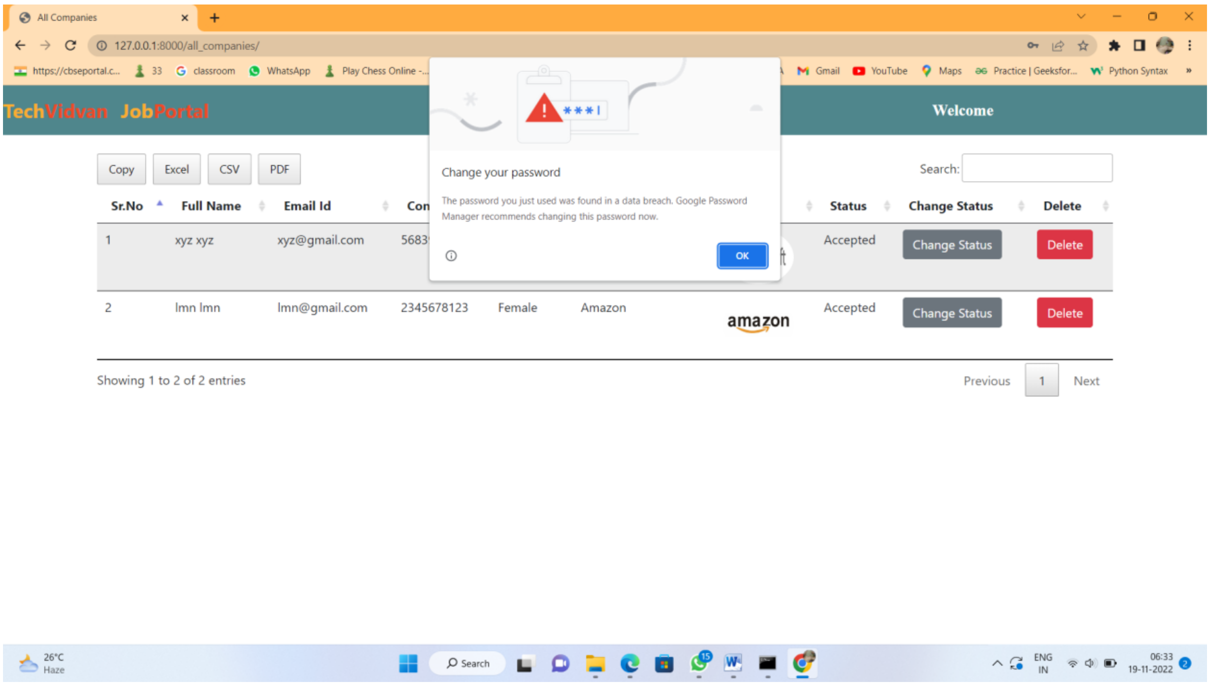


Company Name

Microsoft

Status

Accept Company


All Companies
127.0.0.1:8000/all_companies/
https://cbseportal.c... 33 classroom WhatsApp Play Chess Online -... Gmail YouTube Maps Practice | Geeksfor... Python Syntax
26°C Haze
Search
ENG IN
06:33 19-11-2022

TechVidvan JobPortal

Welcome

Change your password

The password you just used was found in a data breach. Google Password Manager recommends changing this password now.

OK

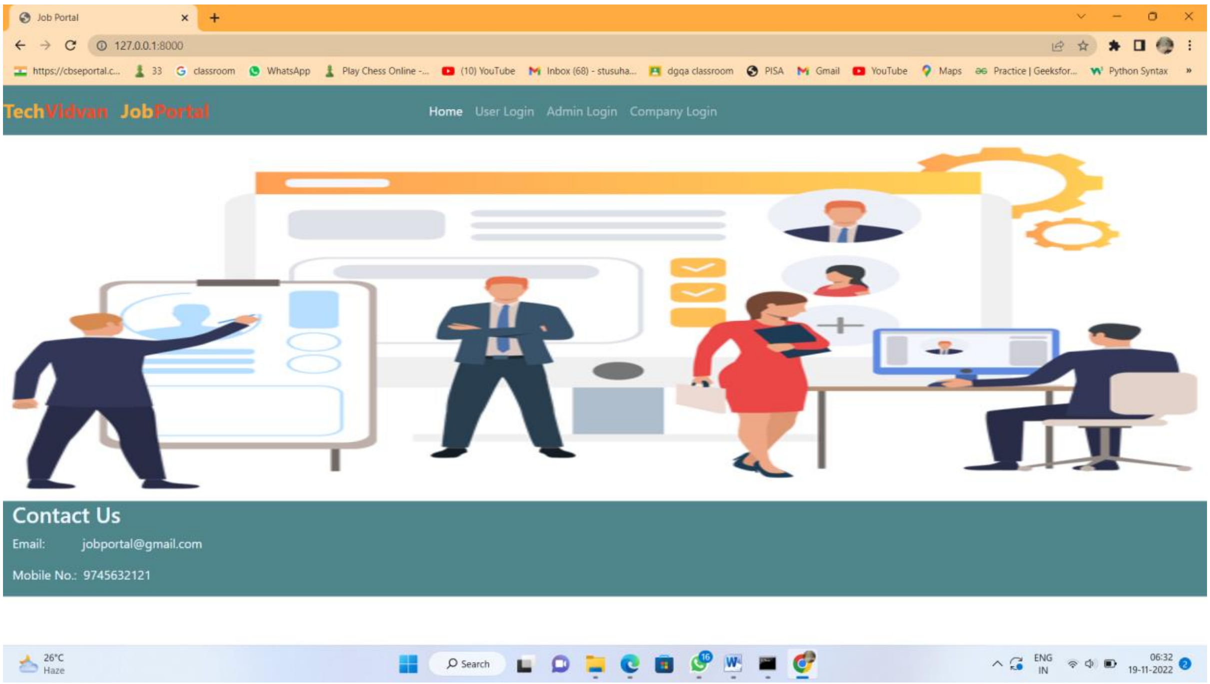
Copy Excel CSV PDF

Search:

Sr.No	Full Name	Email Id	Con				Status	Change Status	Delete
1	xyz xyz	xyz@gmail.com	5683			ft	Accepted	Change Status	Delete
2	lmn lmn	lmn@gmail.com	2345678123	Female	Amazon	amazon	Accepted	Change Status	Delete

Showing 1 to 2 of 2 entries

Previous 1 Next


Job Portal
127.0.0.1:8000
https://cbseportal.c... 33 classroom WhatsApp Play Chess Online -... (10) YouTube Inbox (68) - stusuha... dgqa classroom PISA Gmail YouTube Maps Practice | Geeksfor... Python Syntax
26°C Haze
Search
ENG IN
06:32 19-11-2022

TechVidvan JobPortal

Home User Login Admin Login Company Login

Contact Us

Email: jobportal@gmail.com

Mobile No.: 9745632121

11. CONCLUSION:

In this project, we reviewed the existing literature of numerous publications and proceedings related to job suggestion researches and the hiring process. The issues that the entire e-recruiting platforms faced and our research of the literature have shown us the increased necessity for improving candidate quality and job matching. The recommender system technologies have achieved notable success in a variety of applications and could be effective searching and recommending methods. As a result, there is a significant possibility to use these technologies in the recruitment environment to enhance the quality of the matching jobs.

12. FUTURE SCOPE:

Building more effective AI models and more accurate prediction systems.
Creating a database to make it simple to identify the set of skills required for a certain position.

13. APPENDIX:

Video of the project:
https://drive.google.com/file/d/1Dfx9e31oKs7Ut85IaCPPhSQik5iXvH8P/view?usp=sharing

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-21081-1659771979