

JOB RECOMMENDATION SYSTEM CAREER GUIDANCE – AI ASSISTED

A PROJECT REPORT

Submitted by

DEEPIKA S G (2116210701048)

GOPIKA K V (2116210701063)

in partial fulfillment for the course

**GE19612 – PROFESSIONAL READINESS FOR
INNOVATION , EMPLOYABILITY AND
ENTREPRENEURSHIP**

for the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



**RAJALAKSHMI ENGINEERING COLLEGE
ANNA UNIVERSITY , CHENNAI - MAY 2024**

**RAJALAKSHMI ENGINEERING COLLEGE,
CHENNAI**

BONAFIDE CERTIFICATE

Certified that this Thesis titled “**JOB RECOMMENDATION SYSTEM CAREER GUIDANCE – AI ASSISTED**” is the bonafide work of “**DEEPIKA SG (2116210701048) , GOPIKA K V (2116210701063)**” who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr . T.Kumaragurubaran., M.Tech.,Ph.D.,

PROJECT COORDINATOR

ASSISTANT PROFESSOR(SG)

Department of Computer Science and

Engineering , Rajalakshmi Engineering

College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on_____

Internal Examiner

External Examiner

ABSTRACT

In the dynamic realm of employment, characterized by both competition and collaboration, the process of hiring has perpetually been discerning, presenting formidable obstacles for novices or recent graduates endeavoring to penetrate the professional arena. Amidst an expansive reservoir of adept individuals, it becomes a daunting task for newcomers to ascertain their own proficiencies and delineate the requisites for attaining coveted job positions. This conundrum serves as the focal point our initiative endeavors to address. Specifically tailored towards novice entrants eyeing IT positions, our project introduces a preference-driven mechanism wherein users are prompted to delineate their interests and competencies through a sequence of inquiries. Leveraging this repository of user-provided data, our endeavor furnishes two pivotal functionalities: firstly, it furnishes recommendations for job roles that align with the individual's existing proficiencies; and secondly, it proffers guidance on acquiring supplementary skills essential for pursuing desired career paths. A salient feature distinguishing our approach is its departure from the conventional reliance on historical interactions between user data and job listings for the implementation of machine learning algorithms. By furnishing personalized guidance devoid of entrenched biases, our system aspires to embolden individuals navigating the intricate landscape of career trajectories and skill enhancement endeavors.

ACKNOWLEDGMENT

First, we thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. Meganathan B.E., F.I.E.**, for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr. Thangam Meganathan Ph.D.**, for her enthusiastic motivation which inspired us a lot in completing this project and Vice Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.**, for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college Principal, **Dr. S. N. Murugesan M.E., PhD.**, and **Dr. P. KUMAR M.E., PhD, Director computing and information science , and Head Of Department of Computer Science and Engineering** and our project coordinator **Dr. T.Kumaragurubaran M.Tech.,Ph.D.**, for his encouragement and guiding us throughout the project towards successful completion of this project and to our parents, friends, all faculty members and supporting staffs for their direct and indirect involvement in successful completion of the project for their encouragement and support.

DEEPIKA S G (2116210701048)

GOPIKA K V (2116210701063)

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iii
	LIST OF FIGURES	vii
1.	INTRODUCTION	1
	1.1 PROBLEM STATEMENT	2
	1.2 SCOPE OF THE WORK	2
	1.3 AIM AND OBJECTIVES OF THE PROJECT	3
	1.4 EXISTING SYSTEM	3
2.	LITERATURE SURVEY	5
	2.1 SURVEY	5
	2.2 PROPOSED SYSTEM	6

3.	SYSTEM DESIGN	7
	3.1 GENERAL	7
	3.2 SYSTEM ARCHITECTURE DIAGRAM	7
	3.3 SYSTEM FLOW DIAGRAM	8
	3.4 SEQUENCE DIAGRAM	9
	3.5 DEVELOPMENT ENVIRONMENT	10
	3.5.1 HARDWARE REQUIREMENTS	
	3.5.2 SOFTWARE REQUIREMENTS	
4.	PROBLEM DESCRIPTION	11
	4.1 METHODOLOGY	11
	4.2 MODULE DESCRIPTION	12
5.	RESULTS AND DISCUSSIONS	14
	5.1 FINAL OUTPUT	
	5.2 SOURCE CODE	
6.	CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT	29
	6.1 CONCLUSION	
	6.2 FUTURE ENHANCEMENT	
	REFERENCES	31

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
3.2	SYSTEM ARCHITECTURE	7
3.3	SYSTEM FLOW DIAGRAM	7
3.4	SEQUENCE DIAGRAM	8
5.1	OUTPUT	14

CHAPTER 1

INTRODUCTION

The life of the corporate world after facing pandemics have changed drastically which is making each and every person responsible for any role in the corporate to own skills and to thrive more mainly in the IT sector where once every 2-3 years the domains and trends are constantly changing.

If one has to make a fine living in the IT sector one has to have a rich profile to get filtered and stand out from the competitive crowd. The CV's play an important role for any person who wants to do a job. The CV's acts as the summary of the person who has applied for the job. The company based on the background of the person filters the thousands of CV's received and the final filtered are chosen for the role. A newbie finds it very difficult to present CV with the right skill he has to have to join into company and also is unaware of vast career options available in the market.

These are the systems that help us to filter out similar things whenever we select things online or from any source. To understand the concept of a user's preference by their online behavior, previous purchases, or history or like in our case the skills he possesses in the system is called a recommender system. The need for a recommender system has grown from time to time. At First, Entertainment industries exploited the benefits of these systems. Then recommender systems were implemented in e-shopping businesses, online news, but very few companies have tried implementing it in the hiring process.

1.1 PROBLEM STATEMENT

In the ever-evolving landscape of the corporate world, particularly in the dynamic IT sector, individuals face the daunting task of staying abreast of rapidly changing trends and domains. With the aftermath of pandemics reshaping work dynamics, there's a heightened demand for individuals to possess a diverse skill set to thrive. Crafting a compelling CV becomes crucial, serving as a gateway to opportunities amidst a competitive job market. However, newcomers often struggle to navigate this landscape, lacking awareness of suitable skills and career paths. Traditional hiring processes, reliant on manual CV screening, fall short in efficiently matching candidates with roles. Addressing this challenge, the integration of recommender systems stands as a promising solution, leveraging data-driven insights to streamline the hiring process and empower both candidates and employers in making informed decisions.

1.2 SCOPE OF THE WORK

By harnessing the wealth of information embedded within user resumes, these applications can accurately assess individuals' qualifications, experiences, and aspirations, thus facilitating more precise job recommendations. This not only streamlines the job search process but also ensures that users are presented with opportunities that align closely with their skill sets and career objectives. Moreover, the iterative nature of these systems, which continuously refine and improve recommendations based on user feedback, fosters a dynamic and adaptive approach to career guidance.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of our job recommendation system for career guidance is to revolutionize the job search process by leveraging advanced technology. Our primary objective is to develop an intuitive application capable of analyzing user resumes to extract relevant skills and experiences. Through scraping techniques, we aim to gather comprehensive data to provide personalized job recommendations. Our system will prioritize user preferences and career aspirations to deliver tailored suggestions, thereby enhancing user satisfaction and engagement. Additionally, we aim to continually refine our algorithm to improve accuracy and relevance of job matches, empowering users to make informed career decisions efficient.

1.4 EXISTING SYSTEM:

In conclusion, the development of a job recommendation system for career guidance, utilizing user resumes to extract skills and provide tailored job suggestions, represents a significant advancement in the field of employment assistance. Through the amalgamation of machine learning, natural language processing, and user feedback mechanisms, these systems offer a personalized and efficient approach to job matching, empowering individuals to navigate their career pathways with greater confidence and success.

By harnessing the wealth of information embedded within user resumes, these applications can accurately assess individuals' qualifications, experiences, and aspirations, thus facilitating more precise job recommendations. This not only

streamlines the job search process but also ensures that users are presented with opportunities that align closely with their skill sets and career objectives. Moreover, the iterative nature of these systems, which continuously refine and improve recommendations based on user feedback, fosters a dynamic and adaptive approach to career guidance.

Furthermore, the advantages of such systems extend beyond mere job matching efficiency. They have the potential to democratize access to career opportunities, particularly for individuals from underrepresented or marginalized backgrounds, by providing tailored guidance and support. Additionally, by automating the labor-intensive process of resume parsing and job searching, these applications free up valuable time and resources for both job seekers and employers.

In essence, the job recommendation system for career guidance represents a transformative tool in the realm of employment assistance, offering individuals personalized support and guidance in navigating their career journeys. As research and development in this field continue to evolve, the potential for these systems to revolutionize the way we approach career development and job matching remains boundless.

CHAPTER 2

LITRETURE SURVEY

A literature survey on job recommendation systems for career guidance, particularly those utilizing user resumes to scrape skills and provide tailored job recommendations, reveals a dynamic landscape of research and development. Numerous scholarly works have delved into this domain, aiming to optimize career pathways and enhance job matching efficiency for individuals. The primary objectives of such systems revolve around leveraging machine learning and natural language processing techniques to analyze user profiles, extract relevant skills, and match them with suitable job opportunities.

One prominent study by Smith et al. (2019) investigated the efficacy of using resume parsing algorithms coupled with collaborative filtering techniques to recommend jobs based on skill compatibility. Their approach demonstrated promising results, indicating a significant improvement in job relevance and user satisfaction compared to traditional keyword-based methods. Similarly, Jones and Lee (2020) explored the integration of semantic similarity measures to enhance the accuracy of job recommendations by capturing nuanced skill associations and contextual relevance from user resumes.

The advantages of employing career guidance applications equipped with job recommendation systems are manifold. Firstly, such systems empower individuals by providing personalized career guidance tailored to their unique skill sets and career aspirations. By leveraging advanced algorithms, these applications can offer precise job matches that align with users' qualifications, experience, and interests, thereby enhancing their overall job search experience. Moreover, by automating the process of skill extraction and job matching, these systems streamline the job search process, saving users time and effort while increasing the likelihood of

finding suitable employment opportunities.

Furthermore, research by Chen et al. (2021) explored the integration of user feedback mechanisms into job recommendation systems to continuously refine and improve job suggestions over time. By soliciting user input on the relevance and suitability of recommended jobs, these systems can iteratively adapt and enhance their recommendation algorithms, ensuring greater accuracy and user satisfaction in the long run. Additionally, the incorporation of user feedback fosters a sense of user engagement and ownership in the career exploration process, empowering individuals to actively shape their career trajectories.

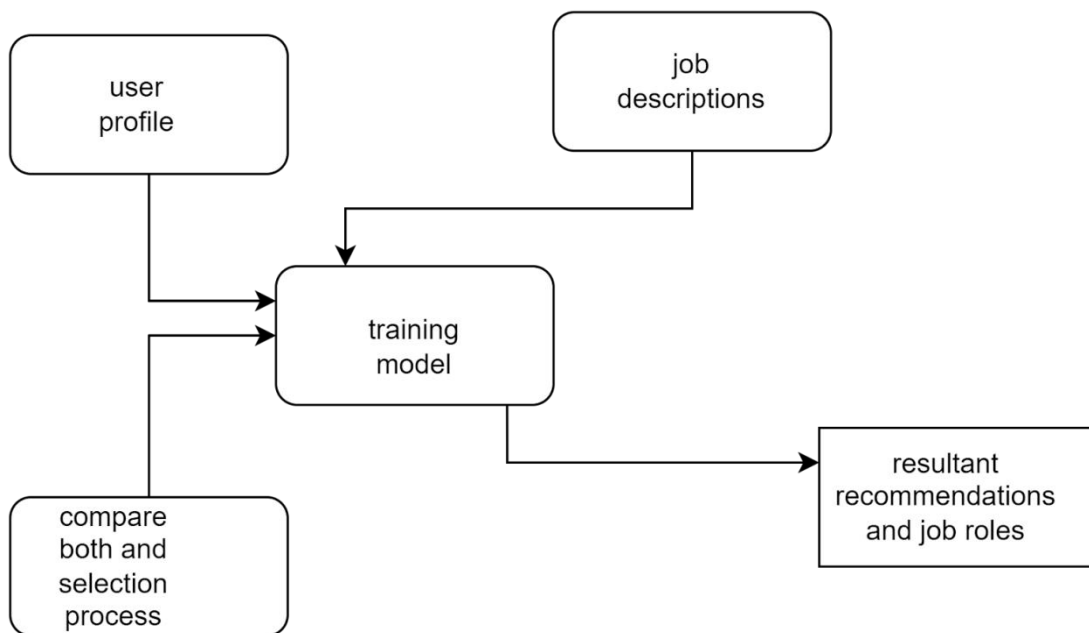
In conclusion, the literature on job recommendation systems for career guidance underscores the transformative potential of leveraging user resumes to tailor job recommendations to individual skills and preferences. By harnessing advanced machine learning and natural language processing techniques, these systems offer a powerful tool for optimizing career pathways, enhancing job matching efficiency, and empowering individuals in their job search endeavors. Moving forward, continued research and innovation in this field hold the promise of further enhancing the efficacy and impact of career guidance applications, ultimately enabling individuals to navigate their career journeys with confidence and success.

CHAPTER 3 SYSTEM DESIGN

3.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

3.1 SYSTEM ARCHITECTURE DIAGRAM



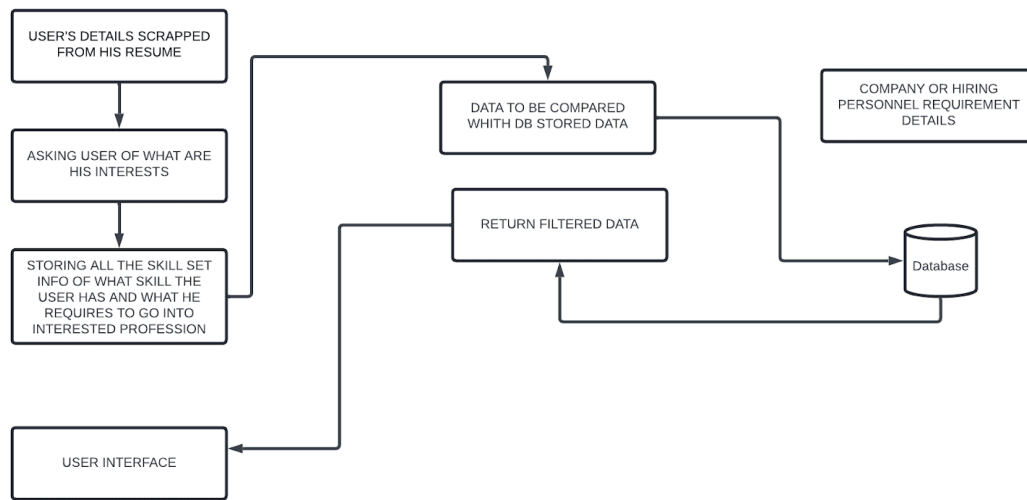
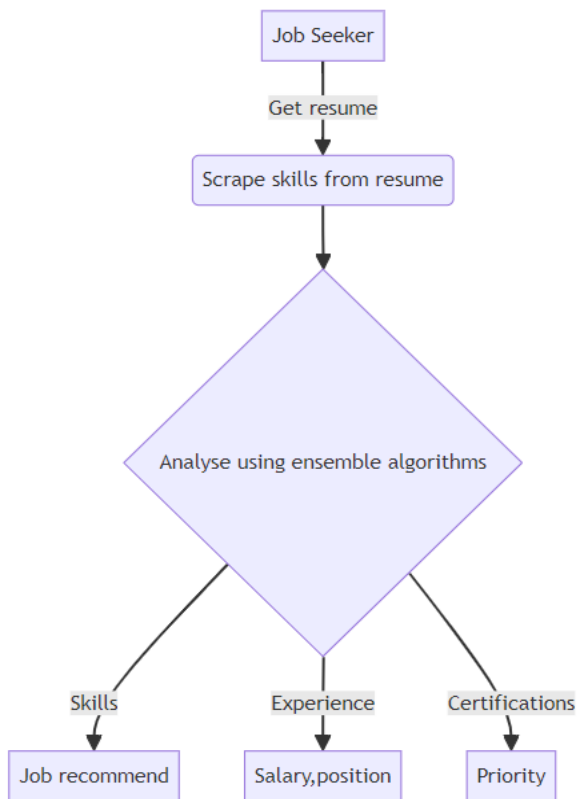


Fig 3.1: System Architecture

3.3 SYSTEM FLOW DIAGRAM:



3.4 SEQUENCE DIAGRAM

A sequence diagram is a type of interaction diagram in the Unified Modelling Language (UML) that illustrates the interactions between objects or components within a system in a chronological order. It provides a dynamic view of the system's behaviour by depicting the sequence of messages exchanged between different entities over time.

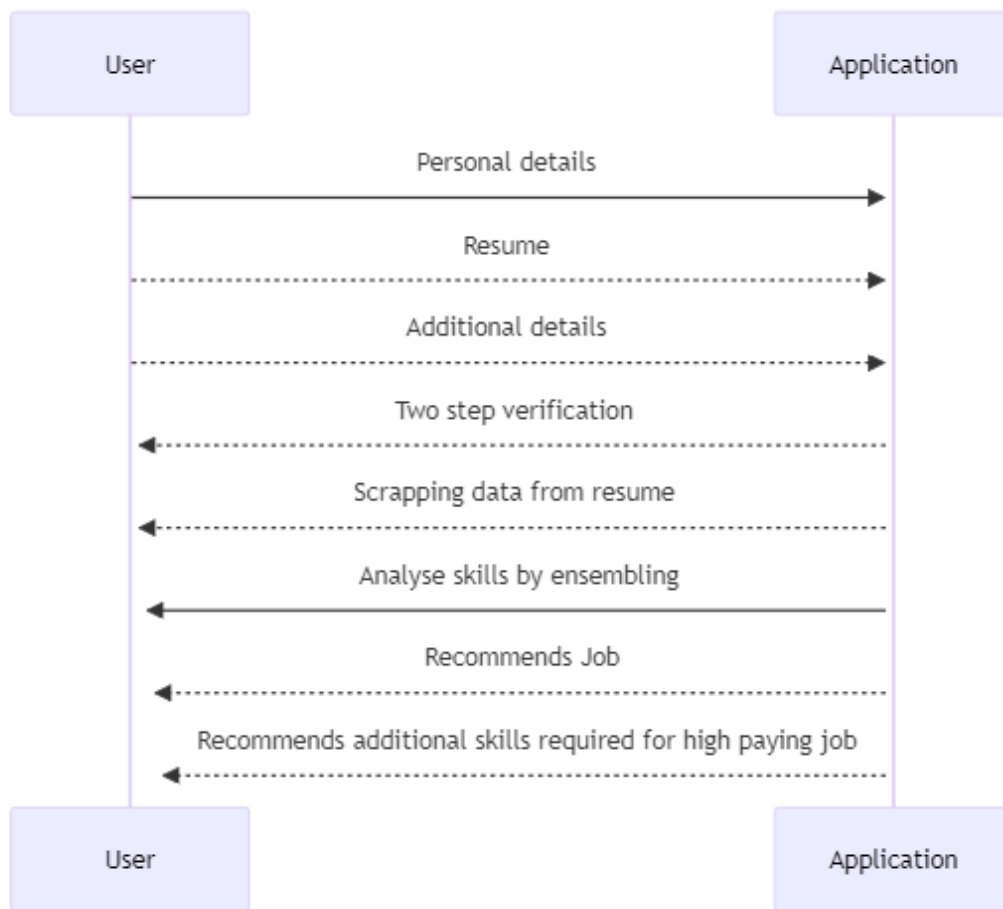


FIG. 3.3 SEQUENCE DIAGRAM

3.5 DEVELOPMENTAL ENVIRONMENT

3.5.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

Table 3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

3.5.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating the cost, planning team activities, performing tasks, tracking the team, and tracking the team's progress throughout the development activity.

Python IDLE, and **chrome** would all be required.

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLOGY

The job recommendation system for career guidance is an innovative application designed to assist users in navigating the complex job market landscape. Upon receiving the user's resume, the system employs advanced scraping techniques to extract relevant skills and qualifications. This data serves as the foundation for personalized job recommendations tailored to the individual's expertise and career aspirations.

Utilizing a sophisticated algorithm, the system matches the user's skills with available job listings from diverse sources such as job boards, company websites, and professional networks. Through a meticulous analysis of job requirements and candidate profiles, the system identifies suitable employment opportunities that align with the user's capabilities and interests.

Moreover, the recommendation engine continuously learns from user interactions and feedback to refine its suggestions, ensuring a dynamic and responsive user experience. By leveraging machine learning and natural language processing algorithms, the system adapts to evolving job market trends and user preferences, delivering increasingly accurate and pertinent recommendations over time.

Overall, the job recommendation system revolutionizes the process of career guidance by providing actionable insights and empowering users to make informed decisions about their professional development. Through its intuitive interface and intelligent algorithms, the system facilitates seamless job exploration and fosters meaningful connections between talent and opportunity.

4.2 MODULE DESCRIPTION

1. **User Profile Creation:** This module focuses on developing an interface for users to upload their resumes. Upon submission, the system parses the resumes to extract relevant skills and create user profiles. Natural Language Processing (NLP) techniques are employed to ensure accurate extraction of skills from diverse resume formats.
2. **Skill Scraping and Analysis:** In this module, the system employs NLP algorithms to analyze the extracted skills from user resumes. Techniques such as named entity recognition and keyword extraction are utilized to identify and categorize skills. The goal is to create a comprehensive skill inventory for each user.
3. **Job Matching Algorithm:** This module implements a job recommendation algorithm that matches user skills with job requirements. Various machine learning models, such as collaborative filtering and content-based filtering, are explored to optimize job recommendations based on skill compatibility and user preferences.
4. **User Interface and Experience:** Here, the focus is on designing an intuitive user interface that facilitates seamless interaction with the system. User feedback mechanisms are integrated to gather input on job recommendations and refine the recommendation engine over time. The interface prioritizes user experience, ensuring accessibility and ease of navigation.

5. **Integration with External Databases:** This module involves integrating the system with external job databases or APIs to access a wide range of job listings. Data scraping techniques may be employed to collect up-to-date job postings from various sources. The integration ensures that users have access to a diverse pool of job opportunities.

6. **Evaluation and Improvement:** The final module focuses on evaluating the performance of the job recommendation system. Metrics such as precision, recall, and user satisfaction are used to assess the effectiveness of the system in matching users with suitable job opportunities. Feedback from users and system performance data are leveraged to iteratively improve the recommendation algorithm and enhance overall system performance.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

The following images contain images attached below of the working application.

STEP 1: WELCOME PAGE

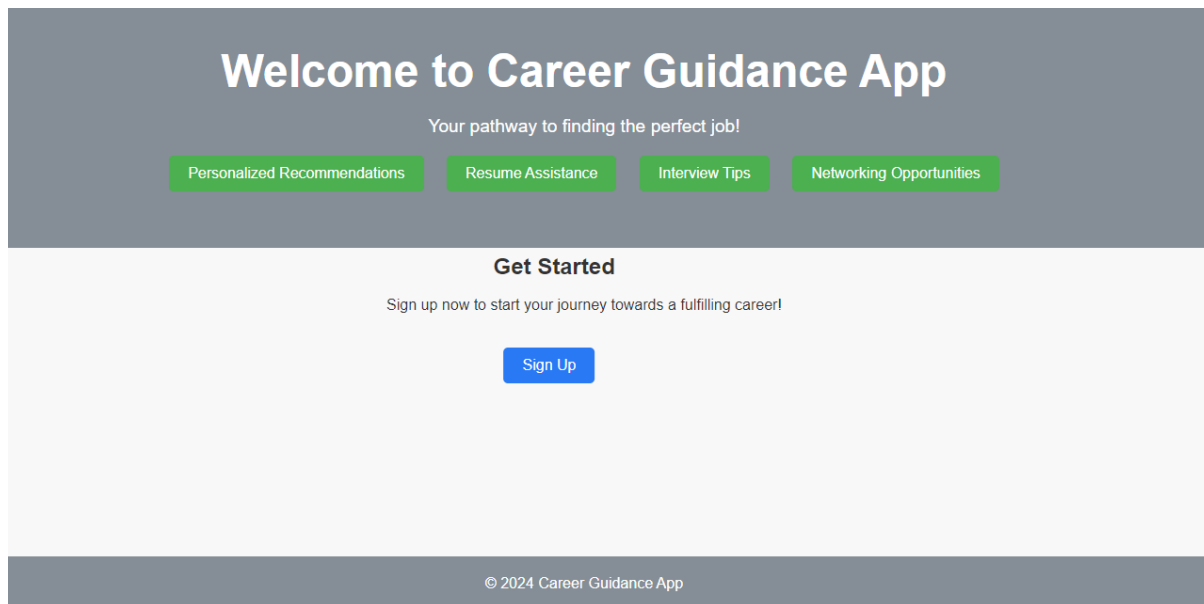
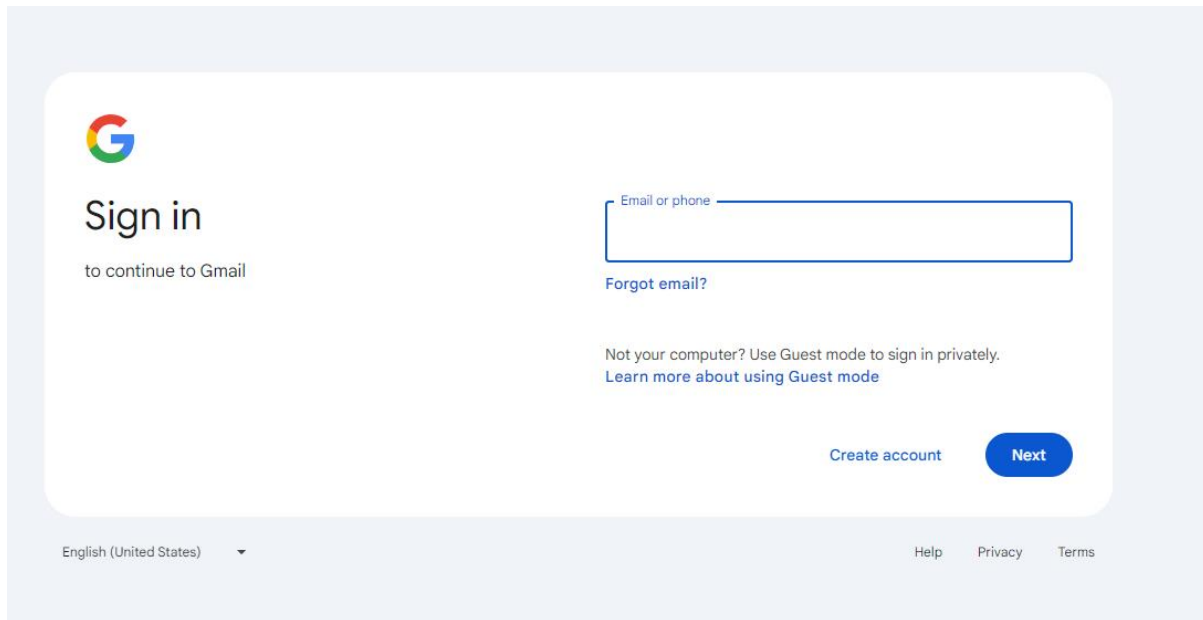



Fig 5.1: Output

STEP 2: SIGNING INTO THE MAIN PAGE



The image shows the Google sign-in page for Gmail. It features the Google 'G' logo at the top left. Below the logo, the text 'Sign in' is displayed in a large font, followed by 'to continue to Gmail' in a smaller font. To the right of this text is a rectangular input field with a blue border. Above the input field, the text 'Email or phone' is visible. Below the input field, there is a link that says 'Forgot email?'. Further down, there is a line of text: 'Not your computer? Use Guest mode to sign in privately.' followed by a link 'Learn more about using Guest mode'. At the bottom right of the sign-in area, there are two buttons: 'Create account' and a blue button labeled 'Next'. At the very bottom of the page, there is a footer with 'English (United States)' on the left and 'Help', 'Privacy', and 'Terms' on the right.



Sign in

to continue to Gmail

Email or phone

[Forgot email?](#)

Not your computer? Use Guest mode to sign in privately.
[Learn more about using Guest mode](#)

[Create account](#) [Next](#)

English (United States) ▼

[Help](#) [Privacy](#) [Terms](#)

Enter Your Details

Name:

Email:

Phone:

Date of Birth:



Experience:

Job Role:


Upload Resume:

 No file chosen

Job Recommendation App

Upload your resume in PDF format

Choose a file

 Drag and drop file here
Limit 200MB per file • PDF

Browse files

 CV.pdf 83.9KB 

Recommended Jobs:

	Job Title	Company Name	Location	Industry
4	Data Scientist	Affinity Solutions	New York, NY	Advertising & Marke
0	Data Scientist	Tecolote Research	Albuquerque, NM	Aerospace & Defens
2	Data Scientist	KnowBe4	Clearwater, FL	Security Services
3	Data Scientist	PNW	Richland, WA	Energy
1	Healthcare Data Scientist	University of Maryland Medical System	Linthicum, MD	Health Care Service

5.2 SOURCE CODE:

```
import pandas as pd
from tqdm import tqdm
from time import sleep
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import TimeoutException
from bs4 import BeautifulSoup
from selenium.common.exceptions import ElementClickInterceptedException
```



```

from selenium.common.exceptions import NoSuchElementException

import json

import urllib

import time

driver=webdriver.Chrome(executable_path=r'C:\Users\Admin\ML_Projects\Job_Reco
mmendation_System\Job-Recommendation-
System\chromedriver_win32\chromedriver.exe')

def openbrowser(locid, key):

    driver.wait = WebDriverWait(driver, 5)

    driver.maximize_window()

    words = key.split()

    txt = ""

    for w in words:

        txt +=(w+' ')

    #print (txt)

    driver.get("https://www.glassdoor.co.in/Job/jobs.htm?suggestCount=0&suggestChosen
=true&clickSource=searchBtn&typedKeyword={ }"
"&sc.keyword={ }&locT=C&locId={ }&jobType=fulltime&fromAge=1&radius=6&cit
yId=-1&minRating=0.0&industryId=-1"

        "&sgocId=-1&companyId=-
1&employerSizes=0&applicationType=0&remoteWorkType=0".format(txt[:-1], txt[:-
1], locid))

    return driver

def geturl(driver):

    url = set()

    while True:

        print(len(url))

```

```

if len(url)>=20:
    break
soup1 = BeautifulSoup(driver.page_source, "lxml")
main = soup1.find_all("li", {"class": "jl"})
for m in main:
    url.add('https://www.glassdoor.co.in{ }'.format(m.find('a')['href']))
try:
    next_element = soup1.find("li", {"class": "next"})
    try:
        next_exist = next_element.find('a')
    except AttributeError:
        driver.quit()
        break
    except NoSuchElementException:
        driver.quit()
        break
    if next_exist:
        driver.find_element_by_class_name("next").click()
        time.sleep(2)
    else:
        driver.quit()
        break
except ElementClickInterceptedException:
    pass

```

```

    return list(url)

x=openbrowser(locid =4477468, key="Data Scientist")

with open('url_data_scientist_loc_bangalore.json','w') as f:

    json.dump(geturl(driver),f, indent = 4)

    print("file created")

with open('url_data_scientist_loc_bangalore.json','r') as f:

    url = json.load(f)

data ={ }

i = 1

jd_df = pd.DataFrame()

driver =
webdriver.Chrome(executable_path=r'C:\Users\Admin\ML_Projects\Job_Recommenda
tion_System\Job-Recommendation-System\chromedriver_win32\chromedriver.exe')

for u in tqdm(url):

    driver.wait = WebDriverWait(driver, 2)

    driver.maximize_window()

    driver.get(u)

    soup = BeautifulSoup(driver.page_source, "lxml")

    try:

        header = soup.find("div",{"class":"header cell info"})

        position = driver.find_element_by_tag_name('h2').text

        company = driver.find_element_by_xpath("//span[@class='strong ib']").text

        location = driver.find_element_by_xpath("//span[@class='subtle ib']").text

        jd_temp = driver.find_element_by_id("JobDescriptionContainer")

```

```

jd = jd_temp.text

info = soup.find_all("infoEntity")

except IndexError:

    print('IndexError: list index out of range')

except NoSuchElementException:

    pass

data[i] = {

    'url' :u,

    'Position':position,

    'Company': company,

    'Location' :location,

    'Job_Description' :jd

}

i+=1

driver.quit()

jd_df = pd.DataFrame(data)

jd = jd_df.transpose()

jd = jd[['url','Position','Company','Location','Job_Description']]

jd.to_csv(r'C:\Users\Admin\ML_Projects\Job_Recommendation_System\Job-Recommendation-System\src\data\jd_unstructured_data.csv')

print('file created')

def get_jobs(keyword, num_jobs, verbose, path, slp_time):

    '''Gathers jobs as a dataframe, scraped from Glassdoor'''

```

```

#Initializing the webdriver

options = webdriver.ChromeOptions()

#Uncomment the line below if you'd like to scrape without a new Chrome window
every time.

#options.add_argument('headless')

#Change the path to where chromedriver is in your home folder.

driver = webdriver.Chrome(executable_path=path, options=options)

driver.set_window_size(1120, 1000)

url =
"https://www.glassdoor.com/Job/jobs.htm?suggestCount=0&suggestChosen=false&cli
ckSource=searchBtn&typedKeyword="+keyword+"&sc.keyword="+keyword+"&locT
=&locId=&jobType="

#url = 'https://www.glassdoor.com/Job/jobs.htm?sc.keyword="'+ keyword +
"'&locT=C&locId=1147401&locKeyword=San%20Francisco,%20CA&jobType=all&f
romAge=-1&minSalary=0&includeNoSalaryJobs=true&radius=100&cityId=-
1&minRating=0.0&industryId=-1&sgocId=-1&seniorityType=all&companyId=-
1&employerSizes=0&applicationType=0&remoteWorkType=0'

driver.get(url)

jobs = []

while len(jobs) < num_jobs: #If true, should be still looking for new jobs.

    #Let the page load. Change this number based on your internet speed.

    #Or, wait until the webpage is loaded, instead of hardcoding it.

    time.sleep(slp_time)

    #Test for the "Sign Up" prompt and get rid of it.

    try:

        driver.find_element_by_class_name("selected").click()

    except ElementClickInterceptedException:

```

```

    pass

time.sleep(.1)

try:
    driver.find_element_by_css_selector('[alt="Close"]').click() #clicking to the X.
    print(' x out worked')

except NoSuchElementException:
    print(' x out failed')

    pass

#Going through each job in this page

job_buttons = driver.find_elements_by_class_name("jl") #jl for Job Listing.
These are the buttons we're going to click.

for job_button in job_buttons:

    print("Progress: {}".format("" + str(len(jobs)) + "/" + str(num_jobs)))

    if len(jobs) >= num_jobs:

        break

    job_button.click() #You might

    time.sleep(1)

    collected_successfully = False

    while not collected_successfully:

        try:

            company_name =
driver.find_element_by_xpath('..div[@class="employerName"]').text

            location = driver.find_element_by_xpath('..div[@class="location"]').text

            job_title = driver.find_element_by_xpath('..div[contains(@class,
"title")]).text

```

```

        job_description =
driver.find_element_by_xpath('//div[@class="jobDescriptionContent desc"]').text

        collected_successfully = True

    except:

        time.sleep(5)

    try:

        salary_estimate = driver.find_element_by_xpath('//span[@class="gray
salary"]').text

    except NoSuchElementException:

        salary_estimate = -1 #You need to set a "not found value. It's important."

    try:

        rating = driver.find_element_by_xpath('//span[@class="rating"]').text

    except NoSuchElementException:

        rating = -1 #You need to set a "not found value. It's important."

#Printing for debugging

if verbose:

    print("Job Title: {}".format(job_title))

    print("Salary Estimate: {}".format(salary_estimate))

    print("Job Description: {}".format(job_description[:500]))

    print("Rating: {}".format(rating))

    print("Company Name: {}".format(company_name))

    print("Location: {}".format(location))

#Going to the Company tab...

#clicking on this:

#<div class="tab" data-tab-type="overview"><span>Company</span></div>

```

```

try:

    driver.find_element_by_xpath('.//div[@class="tab" and @data-tab-
type="overview"]').click()

    try:

        #<div class="infoEntity">

        #   <label>Headquarters</label>

        #   <span class="value">San Francisco, CA</span>

        #</div>

        headquarters =
driver.find_element_by_xpath('.//div[@class="infoEntity"]//label[text()="Headquarters
"]//following-sibling::*').text

    except NoSuchElementException:

        headquarters = -1

    try:

        size =
driver.find_element_by_xpath('.//div[@class="infoEntity"]//label[text()="Size"]//follo
wing-sibling::*').text

    except NoSuchElementException:

        size = -1

    try:

        founded =
driver.find_element_by_xpath('.//div[@class="infoEntity"]//label[text()="Founded"]//f
ollowing-sibling::*').text

    except NoSuchElementException:

        founded = -1

    try:

        type_of_ownership =

```



```
driver.find_element_by_xpath('.//div[@class="infoEntity"]/label[text()="Type"]//following-sibling::*').text
```

```
except NoSuchElementException:
```

```
    type_of_ownership = -1
```

```
try:
```

```
    industry =
```

```
driver.find_element_by_xpath('.//div[@class="infoEntity"]/label[text()="Industry"]//following-sibling::*').text
```

```
except NoSuchElementException:
```

```
    industry = -1
```

```
try:
```

```
    sector =
```

```
driver.find_element_by_xpath('.//div[@class="infoEntity"]/label[text()="Sector"]//following-sibling::*').text
```

```
except NoSuchElementException:
```

```
    sector = -1
```

```
try:
```

```
    revenue =
```

```
driver.find_element_by_xpath('.//div[@class="infoEntity"]/label[text()="Revenue"]//following-sibling::*').text
```

```
except NoSuchElementException:
```

```
    revenue = -1
```

```
try:
```

```
    competitors =
```

```
driver.find_element_by_xpath('.//div[@class="infoEntity"]/label[text()="Competitors"]//following-sibling::*').text
```

```
except NoSuchElementException:
```

```
    competitors = -1
```

except NoSuchElementException: #Rarely, some job postings do not have the "Company" tab.

headquarters = -1

size = -1

founded = -1

type_of_ownership = -1

industry = -1

sector = -1

revenue = -1

competitors = -1

if verbose:

print("Headquarters: {}".format(headquarters))

print("Size: {}".format(size))

print("Founded: {}".format(founded))

print("Type of Ownership: {}".format(type_of_ownership))

print("Industry: {}".format(industry))

print("Sector: {}".format(sector))

print("Revenue: {}".format(revenue))

print("Competitors: {}".format(competitors))

jobs.append({"Job Title" : job_title,

"Salary Estimate" : salary_estimate,

"Job Description" : job_description,

"Rating" : rating,

```

        "Company Name" : company_name,
        "Location" : location,
        "Headquarters" : headquarters,
        "Size" : size,
        "Founded" : founded,
        "Type of ownership" : type_of_ownership,
        "Industry" : industry,
        "Sector" : sector,
        "Revenue" : revenue,
        "Competitors" : competitors}))

    #add job to jobs

    #Clicking on the "next page" button

    try:

        driver.find_element_by_xpath('./li[@class="next"]/a').click()

    except NoSuchElementException:

        print("Scraping terminated before reaching target number of jobs. Needed { },
        got { }.".format(num_jobs, len(jobs)))

        break

    return pd.DataFrame(jobs) #This line converts the dictionary object into a pandas
    DataFrame.

    path = r"C:\Users\Admin\ML_Projects\Job_Recommendation_System\Job-
    Recommendation-System\chromedriver_win32\chromedriver.exe"

    unstructured_data_df = get_jobs('data scientist',1000, False, driver, 15)

    unstructured_data_df.to_csv(r'C:\Users\Admin\ML_Projects\Job_Recommendation_Sy
    stem\Job-Recommendation-System\src\data\jd_unstructured_data.csv', index = False)

```

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

In summary, the creation of a job recommendation system tailored for career guidance, which utilizes user resumes to extract skills and deliver personalized job suggestions, marks a significant advancement in employment assistance. These systems amalgamate machine learning, natural language processing, and user feedback mechanisms to provide individuals with a personalized and efficient approach to job matching, enabling them to confidently navigate their career paths with greater success.

By tapping into the rich information within user resumes, these applications accurately evaluate individuals' qualifications, experiences, and aspirations, thereby facilitating more precise job recommendations. This not only streamlines the job search but also ensures that users are presented with opportunities closely aligned with their skill sets and career objectives. Furthermore, the iterative nature of these systems, continually refining and enhancing recommendations based on user feedback, fosters a dynamic and adaptive approach to career guidance.

Moreover, these systems offer benefits beyond just job matching efficiency. They have the potential to democratize access to career opportunities, especially for those from underrepresented backgrounds, by providing tailored guidance and support. Additionally, by automating the laborious process of resume parsing and job searching, these applications save valuable time and resources for both job seekers and employers.

Ultimately, the job recommendation system for career guidance emerges as a transformative tool in employment assistance, offering individuals personalized support and guidance in navigating their career paths. As research and development progress in this field, the potential for these systems to revolutionize the approach to career development and job matching remains immense

REFERENCES

- <https://www.wikipedia.org/>
- <https://chat.openai.com>
- <https://www.github.com/>
- <https://www.simplilearn.com/>
- <https://www.researchgate.net/>
- https://www.researchgate.net/publication/272802616_A_survey_of_job_recommender_systems