

# **Resume Scanner**

Submitted in partial fulfillment of the requirements of

**NLP Mini Project (CSC702)**

for

**Final Year of Computer Engineering**

By

Kaveya Sivaprakasam 19102A0058

Aditi Shahasane 19102A0054

Pratham Goswami 19102A0061

Yashkumar Charde 19102A0060

Under the Guidance of

Prof. Rasika Ransing

Department of Computer Engineering



**Vidyalankar Institute of Technology**

Wadala(E), Mumbai-400437

**University of Mumbai**

2022-23

# **CERTIFICATE OF APPROVAL**

This is to certify that the project entitled

**“Resume Scanner”**

is a bonafide work of

**Kaveya Sivaprakasam 19102A0058**

**Aditi Shahasane 19102A0054**

**Pratham Goswami 19102A0061**

**Yashkumar Charde 19102A0060**

submitted to the University of Mumbai in partial fulfillment of

**NLP Mini Project (CSC702)**

for

Final Year of Computer Engineering

Guide

Prof. Rasika Ransing

Head of Department

Dr. Sachin Bojewar

Principal

Dr. Sunil Patekar

# Mini Project Report Approval

This project report entitled *Resume Scanner* by

1. *Kaveya Sivaprakasam 19102A0058*
2. *Aditi Shahasane 19102A0054*
3. *Pratham Goswami 19102A0061*
4. *Yashkumar Charde 19102A0060*

is approved for NLP Mini Project (CSC702) for Final Year of Computer Engineering.

Internal Examiner




External Examiner

Date:

Place:

## Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name of student	Roll No.	Signature
Kaveya Sivaprakasam	19102A0058	
Aditi Shahasane	19102A0054	
Pratham Goswami	19102A0061	
Yashkumar Charde	19102A0060	

Date:

Place

## Acknowledgments

This project wouldn't have been possible without the support, assistance, and guidance of a number of people to whom we would like to express our gratitude to. First, we would like to convey our gratitude and regards to our mentor **Prof. Rasika Ransing** for guiding us with his constructive and valuable feedback and for his time and efforts. It was a great privilege to work and study under his guidance.

We would like to extend our heartfelt thanks to our Head of Department, **Dr. Sachin Bojewar** for overseeing this initiative which will, in turn, provide every Vidyalankar student a distinctive competitive edge over others.

We appreciate everyone who spared time from their busy schedules and participated in the survey. Lastly, we are extremely grateful to all those who have contributed and shared their useful insights throughout the entire process and helped us acquire the right direction during this research project.

## **Abstract**

In the present system the candidate has to fill each and every information regarding there resume in a manual form which takes large amount of time and then also the candidates, are not satisfied by the job which the present system prefers according to there skills. Let me tell you a ratio of 5:1 means, If 5 people are getting job than out of that 5, only a single guy will be satisfied by his/her job. Let me tell you an example : If i am a good python developer and particular company hired me and they are making me work on Java so, my python skills are pretty useless. And on the other hand if there is vacant place in a company so according to owner of the company he/she will prefer a best possible candidate for that vacancy. So our system will act as a handshake between this two entities. The company who prefer the best possible candidate and the candidate who prefers the best possible job according to his or her skills and ability.

The problem is that the present are not much flexible and efficient and time saving. It requires candidate, to fill the forms online than also you might not get the genuine information of the candidate. Beside Where our system which saves the time of the candidate by providing to upload there resume in any format preferable to the candidate beside all the information in the resume our system will detect all its activity from the candidate social profile which will give the best candidate for that particular job and candidate will also be satisfied because he will get job in that company which really appreciates candidates skill and ability. On the other hand we are providing same kind of flexibility to the client company.

## Table of Contents

<b>Sr No</b>	<b>Description</b>	<b>Page No</b>
1	Introduction	8
2	Problem Definition	9
3	Literature Survey	10
4	Implementation	11
5	Results	14
6	Conclusion	17
7	References	18

## **Introduction**

Using NLP(Natural Language Processing) and ML(Machine Learning) to match the resumes according to the given constraint, this intelligent system matches the resume of any format according to the given constraints or the following requirement provided by the client company. It will also show by how much percentages the input resume is matching with the required constraints . We will basically take the bulk of input resume from the client company and that client company will also provide the requirement and the constraints according to which the resume should be matched by our system. Beside the information provide by the resume we are going to read the candidates social profiles (like LinkedIn, Github etc) which will give us the more genuine information about that candidate.



## **Problem Definition**

The problem is that the present system are not much flexible and efficient and time saving. It requires candidate, to fill the forms online than also you might not get the genuine information of the candidate. Beside where our system which saves the time of the candidate by providing to upload there resume in any format preferable to the candidate beside all the information in the resume our system will detect all its activity from the candidate social profile which will give the best candidate for that particular job and candidate will also be satisfied because he will get job in that company which really appreciates candidates skill and ability.

# **Literary Survey**

## **Case Study on talent acquisition**

### **1. First Generation Hiring Systems:-**

In this System the Hiring team would publish their vacancies and invite applicants. Methods of publishing were newspaper, television and mouth. The interested candidates would then apply by sending there resumes. These resumes were then received and sorted by the hiring team and shortlisted candidates were called for further rounds of interviews. The whole process would take lot of time and human efforts to find right candidate suitable for their job roles.

### **2. Second Generation Hiring Systems:-**

As the industries have grown, there hiring needs has rapidly grown. To serve this hiring needs certain consultancy units have come into existence. They offered a solution in which the candidate has to upload their information in a particular format and submit it to the agency. Then these agencies would search the candidates based on certain keywords. These agencies were middle level organizations between the candidate and company. These systems were not flexible as the candidate has to upload there resume in a particular formats, and these formats changed from system to system.

### **3. Third Generation Hiring Systems:-**

This is our proposed system, which allow the candidates to upload their resumes in flexible format. These resumes are then analyzed by our system, indexed and stored in a specific format. This makes our search process easy. The analyzing system works on the algorithm that uses Natural Language Processing, sub domain of Artificial Intelligence. It reads the resumes and understands the natural language/format created by the candidate and transforms it into a specific format. This acquired knowledge is stored in the knowledge base.

# Implementation

## Importing libraries

```
!pip install docx2txt
!pip install pdfplumber
!sudo apt install tesseract-ocr
!pip install pytesseract
```

```
import io
import os
import pandas as pd
import docx2txt
from itertools import chain
import pdfplumber
import pytesseract
from PIL import Image
from google.colab import files
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

## Converting file into text

```

def file_to_text(file_path):

    _, file_extension = os.path.splitext(file_path)

    if file_extension == ".docx":
        text = docx2txt.process(file_path).replace("\n", "")
        return text

    elif file_extension == ".pdf":
        text = ""
        with pdfplumber.open(file_path) as pdf:
            num_pages = len(pdf.pages)
            for i in range(num_pages):
                page_content = pdf.pages[i].extract_text().replace("\n", "")
                text += " " + page_content
            return text

    elif file_extension == ".txt":
        with open(file_path, "r") as f:
            text = f.read()
            text = text.replace("\n", "")
            f.close()
            return text

    elif file_extension == ".JPG" or file_extension == ".JPEG" or file_extension == ".PNG":
        image = Image.open(file_path)
        text = pytesseract.image_to_string(image)
        text = text.replace("\n", "")
        return text

    else:
        print("Unsupported Format.")

```

## Uploading job description and resume files

```

job_description = files.upload()
if job_description:
    job_description_path = "/content/" + next(iter(job_description))
    job_description = file_to_text(job_description_path)

```

```

resumes = files.upload()

```

## Pre-processing text data

```

import regex as re
def remove_links(corpus):
    return re.sub(r'http\S+', '', corpus)

```

```

from gensim.utils import tokenize
import nltk
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('omw-1.4')

from nltk.stem import PorterStemmer
from nltk.stem import WordNetLemmatizer
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))

def clean_text(text):
    lemmatizer = WordNetLemmatizer()
    stemmer = PorterStemmer()
    tokens = list(tokenize(text))
    #res = ' '.join([stemmer.stem(t.lower()) for t in tokens if t.lower() not in stop_words])
    res = ' '.join([lemmatizer.lemmatize(t.lower()) for t in tokens if t.lower() not in stop_words])
    if len(res) == 0:
        return ''
    else:
        return res

```

## Importing CountVectorizer library

It is used to transform a given text into a vector based on the frequency (count) of each word that occurs in the entire text.

```
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

```
resume_names = []
similarities = []

for item in chain(resumes.items()):
    resume_name = next(iter(item))
    resume_names.append(resume_name)

cv = CountVectorizer(preprocessor=clean_text)
for name in resume_names:
    path = "/content/" + name
    resume = file_to_text(path)
    resume = remove_links(resume)
    content = [job_description, resume]
    mat = cv.fit_transform(content)
    sim_mat = cosine_similarity(mat)
    sim_per = round(sim_mat[0][1], 4) * 100
    similarities.append(sim_per)
```

Get the match percentage of resume with the given job description

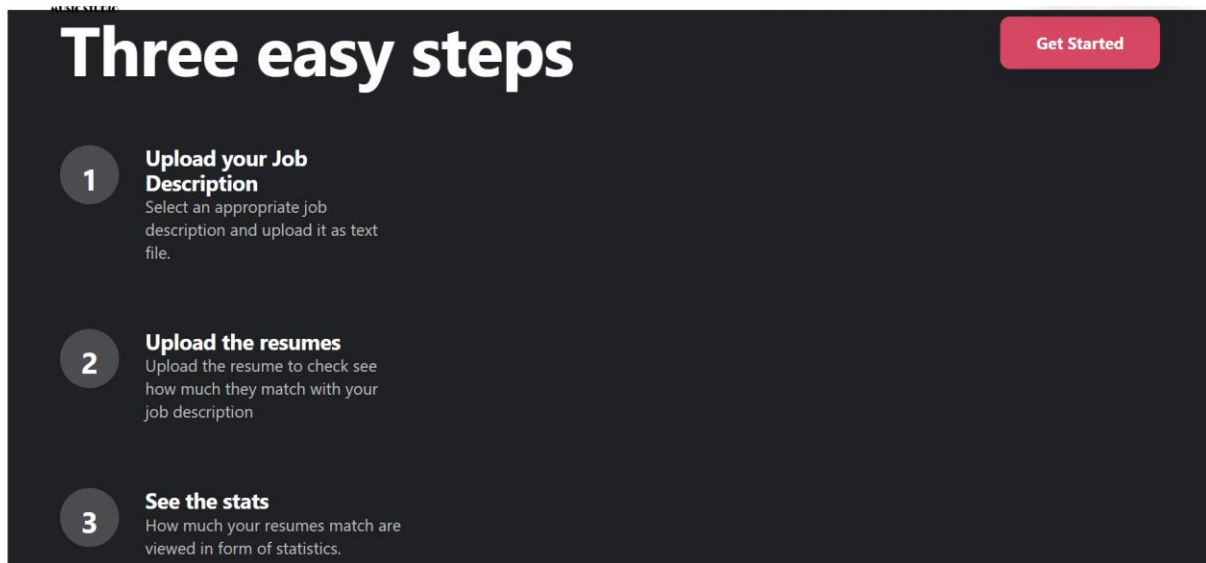
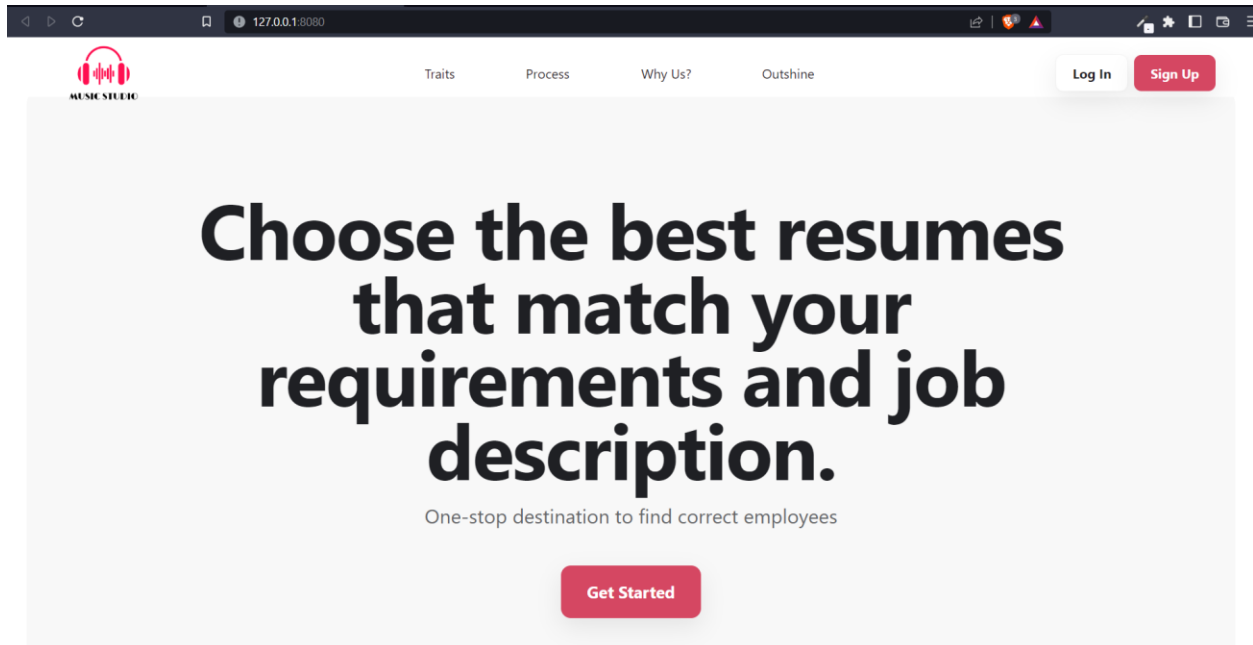
```
data = {"Applicant File": resume_names, "Similarity With Job Description in %": similarities}
data = pd.DataFrame(data)
data.sort_values(by="Similarity With Job Description in %", ascending=True, inplace=True)
print(data)
```

	Applicant File	Similarity With Job Description in %
0	UIUX_Resume1.pdf	2.02
3	IOS1.pdf	10.06
1	android-developer-1559034496.pdf	14.24
4	resume.txt	34.03
2	data-scientist-1559725114.pdf	41.75

## For user interface

We decided to make a website for user interface using Flask as backend and HTML CSS for front end.

Following are the screenshots of the website:







This project has been made for  
[Natural Language Processing Project →](#)

#### Our Process

Research  
Design  
Front End Development  
Back End Development  
Testing

#### Our Team

Kaveya Sivaprakasam  
Pratham Goswami  
Aditi Shahasane  
Yashkumar Charde

#### Our Guides

Prof. Rasika Ransing

## Three easy steps

1

**Upload your Job  
Description**

[Choose File](#)

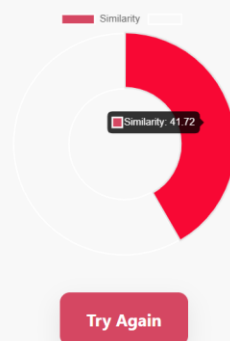
2

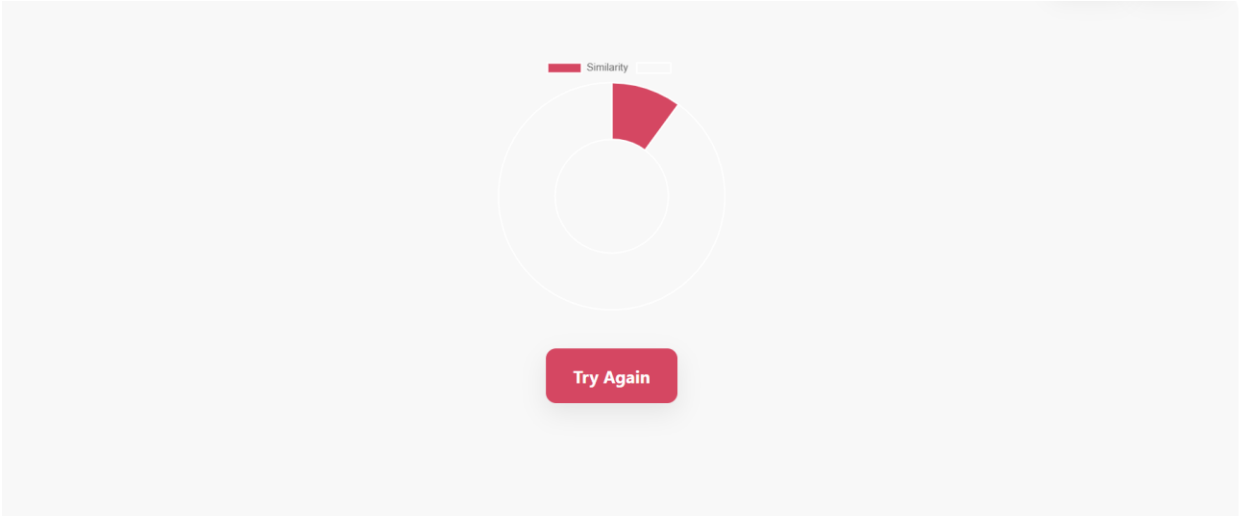
**Upload the resumes**

[Choose File](#)

3

[See the stats](#)





## **Results**

The field of Natural Processing System is gaining momentum especially in this new era of advanced computing. Various fields such as Assessments and Analysis are now taking advantage of this field to optimize the analysis activities. The system not only saves businesses personnel costs but also mitigates the limitations of time and space using the Internet. In this tutorial we have gone through one of the many possible applications of artificial intelligence and text mining on resumes screening programs. Real Applicant Tracking Systems are quite more complex and advanced than the program built in here; they not only scan resumes content but their format too. It is crucial for candidates to know how resume screening systems work to beat them and get their resumes to be viewed by a talent acquisition professional.

Technology is making the job search process easier and harder. Larger pools of applicants and limited opportunities force candidates to write outstanding resumes capable of defeating the “bots”. It is highly recommended for candidates to optimize their resumes keywords to represent their soft and hard skills and avoid including buzzwords.

## **Conclusion**

Our system will provide better and efficient solution to current hiring process. This will provide potential candidate to the organization and the candidate will successfully be placed in an organization which appreciate his/her skillset and ability. The application can be extended further to other domains like Telecom, Healthcare, Ecommerce, and public sector jobs. This software can automate the process of job hiring. In colleges this can be used in a larger scale since it would be easier for the Placement Cell to assign each student to the right company.

## References

[1] IEICE TRANS. INF. & SYST., VOL.E94–D, NO.10 OCTOBER 2011 Special Section on Information-Based Induction Sciences and Machine Learning A Short Introduction to Learning to Rank, Hang LI

[2] Identifying “best” applicants in recruiting using data envelopment analysis Sharon A. Johnson, JoeZhu.

<http://www.sciencedirect.com/science/article/pii/S0038012102000484>

[3] Referenced Links:

Jessica Simko , “How Hiring Managers Make Decisions”

<http://www.careerealism.com/hiring-managers-decisions/>

Vinayak Joglekar , “Ranking Resumes using MachineLearning”

<https://vinayakjoglekar.wordpress.com/2014/06/24/ranking-resumes-using-machine>

Peter Gold , “Artificial Intelligence Recruiting”

<https://www.linkedin.com/pulse/artificial-intelligence-recruiting-peter-gold>

Turbo Ricruit , “Automated Application Processing”, “Better candidate experience”, “Matching Job Descriptions to Resumes”

<http://www.turborecruit.com.au/benefits-of-artificial-intelligence-for-recruitment/>