

A Project Report
On
RESUME ANALYSIS USING NLP – RESUMET
Submitted in partial fulfilment of the
Requirements for the award of the Degree of
MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
By
JAMEEL SHAIKH
SEAT NUMBER: 4135647
Under the esteemed guidance of
Prof. Sheetal Mathew



DEPARTMENT OF INFORMATION TECHNOLOGY
GURU NANAK COLLEGE OF ARTS, SCIENCE AND
COMMERCE

(Affiliated to University of Mumbai)

MAHARASHTRA, MUMBAI - 400031

2022 – 2023

RESUME ANALYSIS USING NLP

RESUMET

A Project Report
On
RESUME ANALYSIS USING NLP – RESUMET
Submitted in partial fulfilment of the
Requirements for the award of the Degree of
MASTER OF SCIENCE (INFORMATION TECHNOLOGY)
By
JAMEEL SHAIKH
SEAT NUMBER: 4135647
Under the esteemed guidance of
Prof. Sheetal Mathew



DEPARTMENT OF INFORMATION TECHNOLOGY
GURU NANAK COLLEGE OF ARTS, SCIENCE AND
COMMERCE

(Affiliated to University of Mumbai)

MAHARASHTRA, MUMBAI - 400031

2022 – 2023

GURU NANAK COLLEGE OF ARTS, SCIENCE AND COMMERCE

(Affiliated to University of Mumbai)
MAHARASHTRA, MUMBAI - 400031

2022 – 2023

DEPARTMENT OF INFORMATION TECHNOLOGY



CERTIFICATE

This is to certify that the project entitled, "**Resume Analysis Using NLP - Resumet**" is bonafied work of Mr. **Jameel Shaikh** bearing Seat No **4135647** submitted in partial fulfilment of the requirements for the award of degree of **MASTER OF SCIENCE** in **INFORMATION TECHNOLOGY** from University of Mumbai.

Project Guide

Internal Examiner

Date:

Co – Ordinator

External Examiner

College Stamp

DECLARATION

I hereby declare that the project entitled, "**Resume Analysis Using NLP - Resumet**" done at place where the project is done, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree of **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as project as part of our curriculum.

Name and Signature of the Student

ACKNOWLEDGEMENT

I sincerely wish to thank my project guide **Prof. Sheetal Mathew** for his ever encouraging and inspiring guidance helped us to make my project a success. My project guide made me endure with her expert guidance, kind advice and timely motivation which helped us to determine about my project.

I also express our deep gratitude to our honorable Principal **Mrs. Pushpinder Bhatia** for giving us an opportunity to study and pursue a career in Information Technology

ABSTRACT

Agencies and various high-level firms must deal with many new jobs seeking people with various resumes. However, managing large amounts of text data and selecting the best-fit candidate is more difficult and time-consuming. This paper provides an overview of an ongoing Information Extraction System project that helps recruiters in identifying the best candidate by extracting relevant information from the resume.

This project presents a system that uses Natural Language Processing (NLP) techniques to extract minute data from a resume, such as education, experience, skills, and experience. The recruiting process is made easier and more efficient by parsing the resume. The proposed system is made up of three modules: an administration management system, File upload and parser system, and an information extraction system. The administrator will upload the applicant's resume into the system, and the relevant information will be extracted in a structured format. Using the parsed information from the Resume, HR can select the best candidate for the job based on the company's needs.



Resume Analyser

RESUME ANALYSIS USING NLP RESUMET

By:

Jameel Shaikh

Contents

1. Introduction	1
1.1. Problem Statement.....	2
1.2. Implementation of module	2
1.2.1. Reason behind implementing NLP	2
1.2.2. Work Flow of NLP	3
1.3. Aims	4
1.4. Objectives	4
1.5. Artefacts	4
1.6. Scope and Limitations of the project.....	6
2. Review of Literature.....	6
3. Project Methodology	7
3.1 Scrum Methodology	7
3.1.1. Reason behind Selection.....	8
3.2. Initial Gantt Chart.....	9
3.3 Additional info.....	9
3.3.1 Resources	9
4. Technology and Tools	10
5. Diagrams	12

5.1. Activity Diagram	12
5.2. Use Case Diagram	13
5.3. Sequence Diagram.....	14
5.4. Screenshots	16
5.5. Design Process.....	21
5.6. Development.....	21
5.7. Module Implementation	22
5.7.1 Important Necessary Libraries and Module.....	22
5.7.2. Code	24
6. Conclusion	33
7. Critical Evaluation.....	33
7.1. Final Report	33
7.2. Finding and process	33
7.3. System	33
7.4. Future Work.....	34
7.5. Self-reflection	34
8. References	35

1. Introduction

Resume Analyzer is a system for determining whether a candidate is qualified for a role based on his or her education, experience, skills, and other information captured on their resume. It also helps candidates to prepare a better resume and themselves for the interview.

With the help of keyword matching algorithm, it identifies the designation of the candidate and based on that, it shows the level at which the candidate is and recommends the candidate skills required for that designation.

As large numbers of people apply for the job, analyzing each resume is difficult work. Also scheduling the interviews as per the sorting is a tedious job. We are attempting to reduce efforts on candidate's as well as company's side.

Structured Data (Name, Email_ID, Resume_Score, Timestamp, Pages, predicted_Field, User_Level, Skills, recommended_Skills, recommended_Courses)

Name	
Email_ID	
Resume_Score	
Timestamp	
Pages	
predicted_Field	
User_Level	
Skills	
recommended_Skills	
recommended_Courses	

1.1. Problem Statement

Nowadays while applying for any job vacancy applicant needs to fill the company's application form and upload resume on company's website. The application form, in general, is very lengthy and time consuming to design as well. The applicant needs fill all the details in the form which is very tedious and time-consuming process. The information related to applicant is redundant as data from resume is same as that extracted from the form. The space is wasted in saving redundant data which is not required.

1.2. Implementation of module

By collecting the relevant information from the resume, a **natural language processing** technique is employed to create a Hire ability system. Different NLP libraries, such as NLTK and Spacy, are utilized for extraction. Natural language networks are usually taught as **unsupervised** techniques, which implies that no previous tagging or labeling is done before the model is trained.

Natural Language Processing is a branch of AI that integrates languages and computer science to study the patterns and architecture of language. It helps to develop intelligent systems that are based on machine learning and NLP algorithms which can read, analyze, and extract meaning from text and voice.

1.2.1. Reason behind implementing NLP

The key reason I chose NLP for Hireability is that it can handle massive volumes of data in seconds or minutes that would take days or weeks to analyze manually. NLP technologies can instantly scale up or down to match demand, allowing us to have as much or as little processing capacity as per requirement.

Aside from that, humans are prone to errors or may have personal biases that might distort the findings while conducting repeated jobs like examining resumes one by one and other textual data. In this case, NLP-powered solutions can be taught to understand company's need and requirements in only a few steps. So, once they're up and going, they perform better in terms of accuracy (Wolff, 2020).

1.2.2. Work Flow of NLP

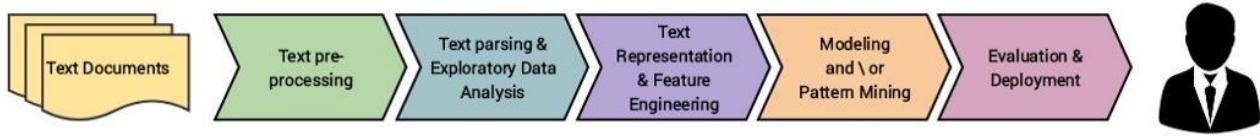


Figure 1.2.2: Generic Workflow of NLP

Natural language processing comprises a wide variety of methods for analyzing human language, based on machine learning techniques as well as rules-based and computational approaches. Tokenization, lemmatization and stemming, parsing, part-of-speech tagging, language identification are some basic NLP tasks. NLP tasks, in general, break down the language into smaller, essential components, attempt to comprehend links between the pieces, and then examine how the components combine to form meaning (C, 2020).

The procedures given above represent the standard workflow for an NLP task.

The initial stage is generally text wrangling and pre-processing on the collection of documents. Then there's parsing and some basic exploratory data analysis. The next stage is to represent text using word embeddings and then do feature engineering. Following that, we must select a model based on whether we are dealing with a supervised or unsupervised learning scenario. The final step of any ML workflow is model testing and deployment. The early processes of text pre-processing and EDA.

1.3. Aims

- To take the help of the cutting-edge and latest NLP technology to enhance their business processes.
- To extract the required information about candidates without having to go through each resume manually.
- To replace slow and expensive human processing of resumes with extremely fast and cost-effective software.
- To help candidates, to write better resume and prepare them for the interview.

1.4. Objectives

- For information extraction, NLP model will be configured and reconfigured.
- A system for uploading resumes will be developed.
- The unstructured data will be transformed into structured form.

1.5. Artefacts

A subsystem is a fundamental component of a larger system. Dividing the system into subsystems will aid in the development of a more advanced version. The following are the three key artefacts that will be developed for this project in order to create a functional model.

1. Admin management system

This sub-system is one of the most secretive aspects of the project. As it is a carefully guarded part of the project, only the administrator can access using the admin email ID and password. After the admin logged in, there'll be given the option to upload the resume. Following the submission of the resume, there will have two sub-models: resume parsing and information extraction.

The following is a brief list of how this artefact will be created and the tools required to do it.

Methodology: SCRUM

Design Tool: Figma

Frontend: HTML, CSS, JavaScript

Backend: Flask/Django

IDE: VsCode/ Sublime text

Version Control: Git

Browser: Google Chrome or Mozilla (Alternate)

2. File upload and Parsing system

This system is at the center of the overall project. In this sub system the admin will upload the resume and go for further processing. The parser subsystem will use different libraries to transform the submitted unstructured resume to a structured format. This will make it much easier to examine, analyze, and grasp the data.

The following is a brief list of how this artefact will be created and the tools required to do it.

Methodology: SCRUM

Data Requirement: dataset collected from Kaggle

Data analysis and evaluation: Jupyter Notebook/VsCode

IDE: VsCode/Sublime text

Version Control: Git

Browser: Google Chrome or Mozilla

Build Tools: Pip

3. Information extraction system

After parsing the data, we employ the NLTK, Spacy phrase matchers, regex to extract the essential information. The required extracted data will be saved in JSON format.

Finally, the extracted data or dump JSON file is stored in a MySQL database for further usage if it is needed.

The following is a brief list of how this artefact will be created and the tools required to do it.

Methodology: SCRUM Design Tool: Figma **Programming Language:** Python 3 Technologies

Libraries: NLTK, Spacy

Data analysis and evaluation: Jupyter Notebook/VsCode

IDE: VsCode

Version Control: Git **Build Tools:** Pip

1.6. Scope and Limitations of the project

A purpose system can help in resolving the challenge of obtaining useful information from a resume in a structured format. By resolving this issue, recruiters will be able to save hours each day by eliminating manual resume screening. Bias in hiring is still prevalent, thus this method may also address the bias hiring process and strengthen a non-bias policy.

Purpose system is not able to solve the complex network issue such as:

- Excessive web traffic can significantly slow down or restrict access to a website entirely. This occurs when the server receives more file requests than it can handle.
- Latency issues

2. Review of literature:

Agencies and different high-level companies have to deal with an extreme number of new jobs seeking employees with different resumes. However, looking after those large numbers of text data and filtering out the needed candidates is a burden on the brain and more time consuming.

Therefore, the essence of this literature review is on studying resumes in different formats such as single-column resumes, double-column resumes with extension.pdf,.docx, and how the suggested

Information Extraction System converts that unstructured information into structured layout through Parsing.

Accordingly, this review also helps to understand and apply several in-use and well recognized algorithms currently being used in industries to reduce human labor. Depending upon the Company's preference to hire employees, the Extraction System will manage the gathered information with more readability and organized data forms. Furthermore, the analysis of various machine learning algorithms and natural language processing techniques would be equally carried out along with their proper implementation and evaluation. The reviews from multiple research publications and journals are included below.

3. Project Methodology

A collection of guiding concepts and practices for planning, managing, and executing projects is known as project management methodology. The project management approach that we select affects how work is prioritized and finished. Here, are a few well-known project management approaches in use today.

1. Waterfall methodology
2. Agile methodology
3. Kanban methodology

Agile is a software development process that is continuous and responsive. High levels of communication and cooperation, rapid and convenient reactions to change, adaptable planning, and continuous improvement are all characteristics of Agile development. Scrum, Kanban are just a couple of the techniques and sub-frameworks that have resulted in the emergence of the agile project management concept. Each framework has its own set of advantages and disadvantages.

3.1 Scrum Methodology

To do Hire ability project as yearlong project I have Chosen scrum. It is the most widely used Agile methods framework due to its ease of use and efficiency.

3.1.1. Reason behind Selection

The major reason I chose this method is:

- It is a transparent approach to project management.
- The tasks and the progress of each task are constantly visible during the sprint.
- It is easy to learn and use.
- It even reduces time taken to get product to market.

Scrum divides work into small cycles known as "sprints," which typically run approximately one to four weeks. For each sprint iteration, work is taken from the backlog. For the course of the sprint, small teams are supervised by a Scrum Master, after which they analyze their work in a "sprint retrospective" and make any required changes before beginning the next sprint. Testing thoroughly, paying close attention to detail, and learning from prior Sprints all contribute to a high level of quality (teamwork, 2021).

Sprint Backlog					
Task Number	Description of the task	Start date	End date	Sprint Goal	Sprint
1	Project Planning	12/18/2021	12/25/2021	Requirements gathering	1
2	Wireframe	12/25/2021	12/30/2021	Draw wireframe	
3	UML Diagram	1/1/2022	12/30/2021	Draw UML Diagram	
4	UI/UX Design	1/1/2022	1/8/2022	Draw Design	
5	Machine Learning Model	1/11/2022	2/10/2022	Build, train and Test MI Model	
6	Frontend Development	2/11/2022	2/25/2022	development of frontend	2
7	Backend Development	2/25/2022	03/12/2022	connecting system using backend coding	
8	Model Integrate	3/13/2022	04/10/2022	Adding build MI model to system	
9	Testing	04/11/2022	4/15/2022	Test the system	
10	Documentation	4/16/2022	4/26/2022	Final Report Writing	

Figure 3.1: Sprint Backlog

3.2. Initial GANTT Chart



Figure 3.2: Initial Gantt Chart

3.3 Additional info

3.3.1 Resources

The resources necessary for the entire project's development are listed below.

Hardware Requirements:

- PC with 128 GB of SSD, Python Development Environment and a stable internet connection
- Windows and Linux for testing
- Figma for UI/UX design

Software Requirements:

- Figma for logo design
- Google Chrome or Mozilla (Alternate) for research and debugging
- Visual Studio Code for Python Development
- Git for version control
- Google Drive for the document (Word/ PDF) controls
- Microsoft Visio/draw.io for creating WBS and other diagrams
- Excel for creating Gantt Chart
- MySQL for storing data if needed
- Pip for installing python packages
- NLTK and SpaCy for resume parsing and extraction
- Flask for developing web application
- Jupyter Notebook or Vscode for evaluating and exploring data

4. Technology and Tools

During the project's development, a variety of tools were used. The list of Software tools used in the project is divided into two categories: hardware and software tools.

1. Programming Language:

For this project, I employed the **Python programming language**. The major reason for using Python is because of its simple syntax, which makes it versatile, simple to use, and quick to develop. Another reason to prefer Python is that it can be used for web development and is ideal for ML and AI applications.

2. Framework:

When I began exploring frameworks, **Django** was at the top of the list. Flask, Laravel, Ruby on Rails, and CakePHP were among the other alternatives, but according to research and evaluations, the Django framework was more reliable, fast-simple, and, most importantly, perfect for any web application project. It's a Python web frame. As a result, I began to focus on Django for the project's backend by watching videos, reading manuals, and so on.

3. Web Technologies:

For frontend development, **HTML, CSS, JavaScript**, and Bootstrap were chosen. HTML provides the foundation for a website's structure, which is then improved and updated by CSS and JavaScript. CSS is used to manage the appearance, style, and layout of a page, whereas JavaScript is used to control the behavior of certain components.

4. IDE:

For the coding side, I used **vscode**. As, VS Coding is a highly capable code editor with built in git integration. It attracted to me since it is simple to use, write code in, add files, and most importantly, it provides syntax highlighting and bracket-matching.

5. Version Control:

When working on any project, using Version Control is essential since it allows individuals to rapidly track changes made to the project and ensures that everyone is working on the most recent version. So, using a variety of commands, I uploaded my project to **Github** and committed it every time I made a change to the code.

6. Browser:

I choose **Google Chrome** as my browser since it is simple to use and is faster. It also loads web pages with a single click and can open many tabs at once.

5. Diagrams

5.1. Activity Diagram

The behavior of a system is depicted in an activity diagram. The control flow from a start point to a completion point is represented in an activity diagram, which shows the numerous decision routes that exist while the activity is being performed. The activity diagram for the admin management system is shown below (geeksforgeeks, 2018).

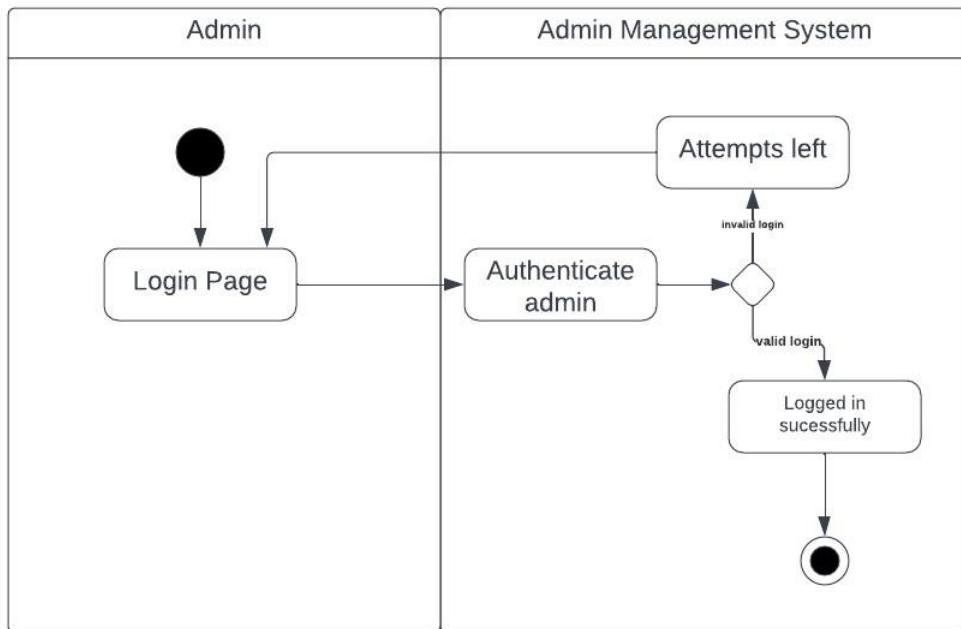


Figure 5.1: Activity Diagram of admin management system

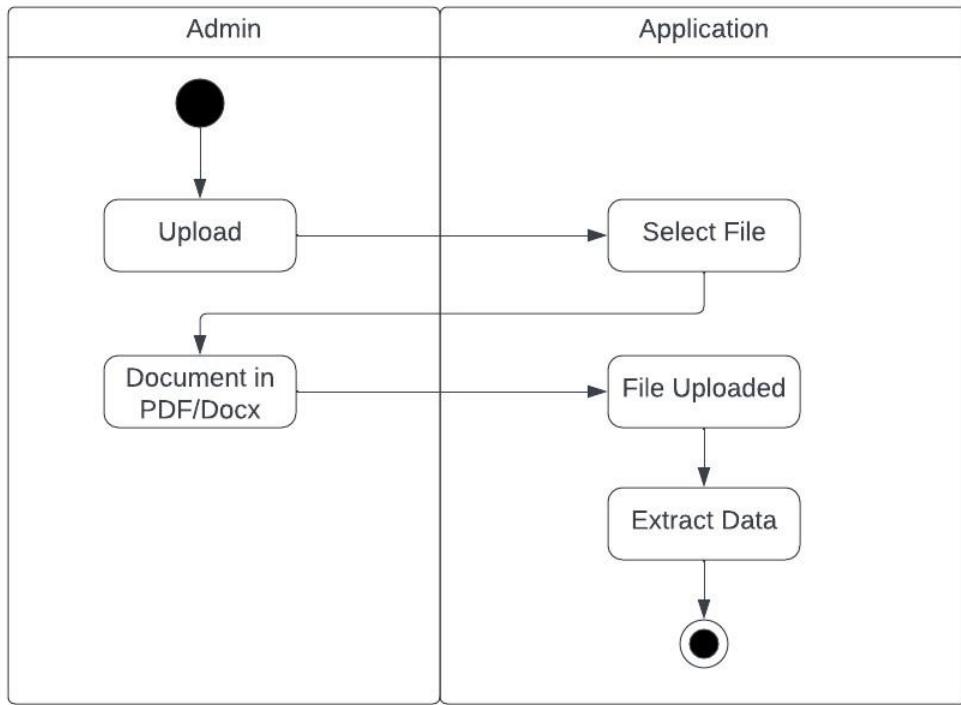
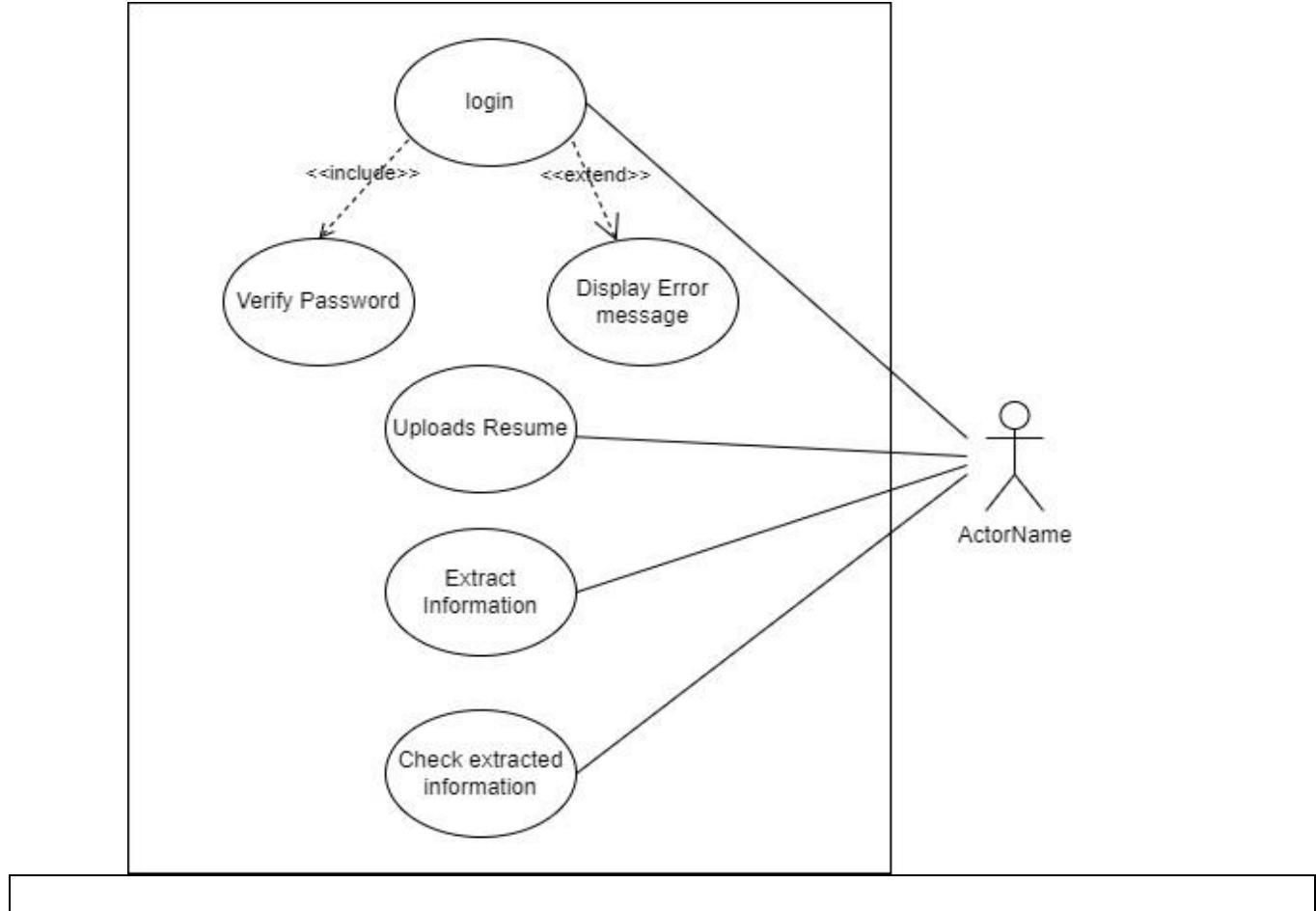


Figure 5.2 activity diagram of ~~Figure 5.2 activity diagram of~~ parsing system

5.2. Use Case Diagram

The interactions between the system and its actors are mostly identified with the use case diagram. In use-case diagrams, use cases and actors define what the system does and how the actors interact with it. There is just one user in the diagram below, and that is the admin. Admin can log in, navigate to the upload page, upload their resume, and view the extracted data (ibm, 2021).



5.3. Sequence Diagram

A sequence diagram displays the order in which items interact in a sequential way. It shows how and in what sequence the components of a system work together. The sequence diagram for the admin management system is shown below (geeksforgeeks, 2018).

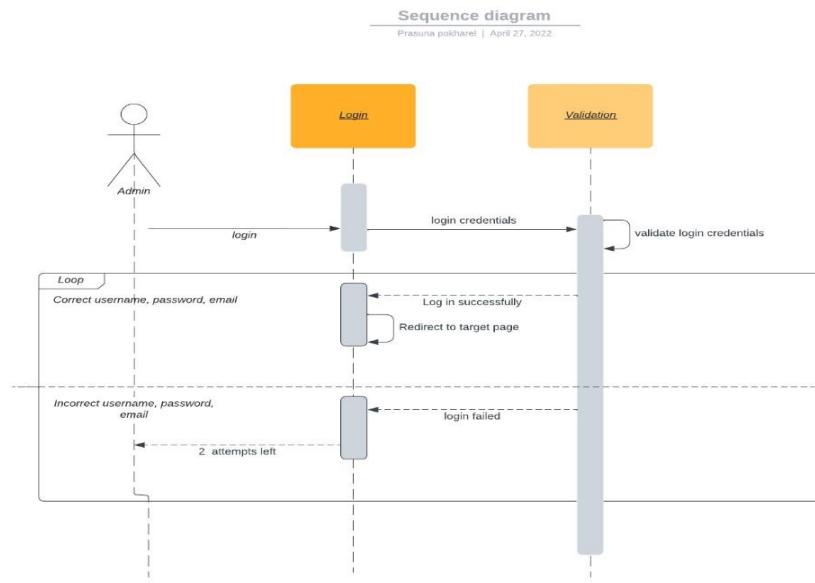


Figure 6.1: Admin Management System

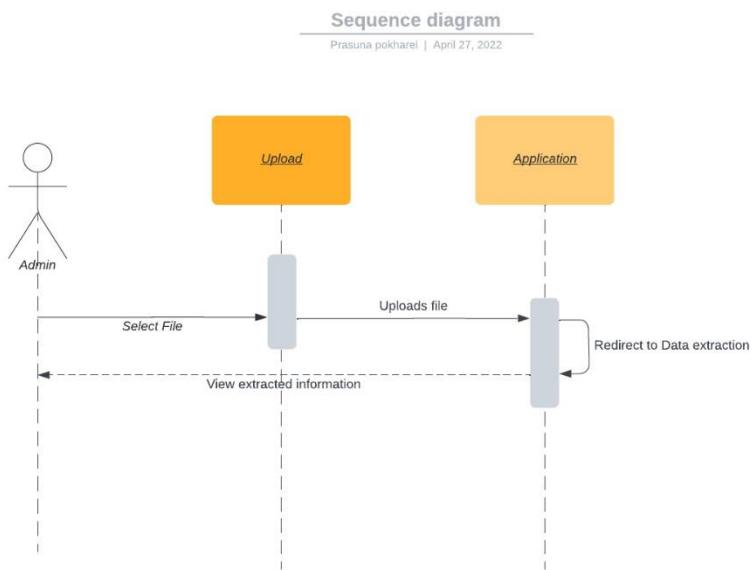
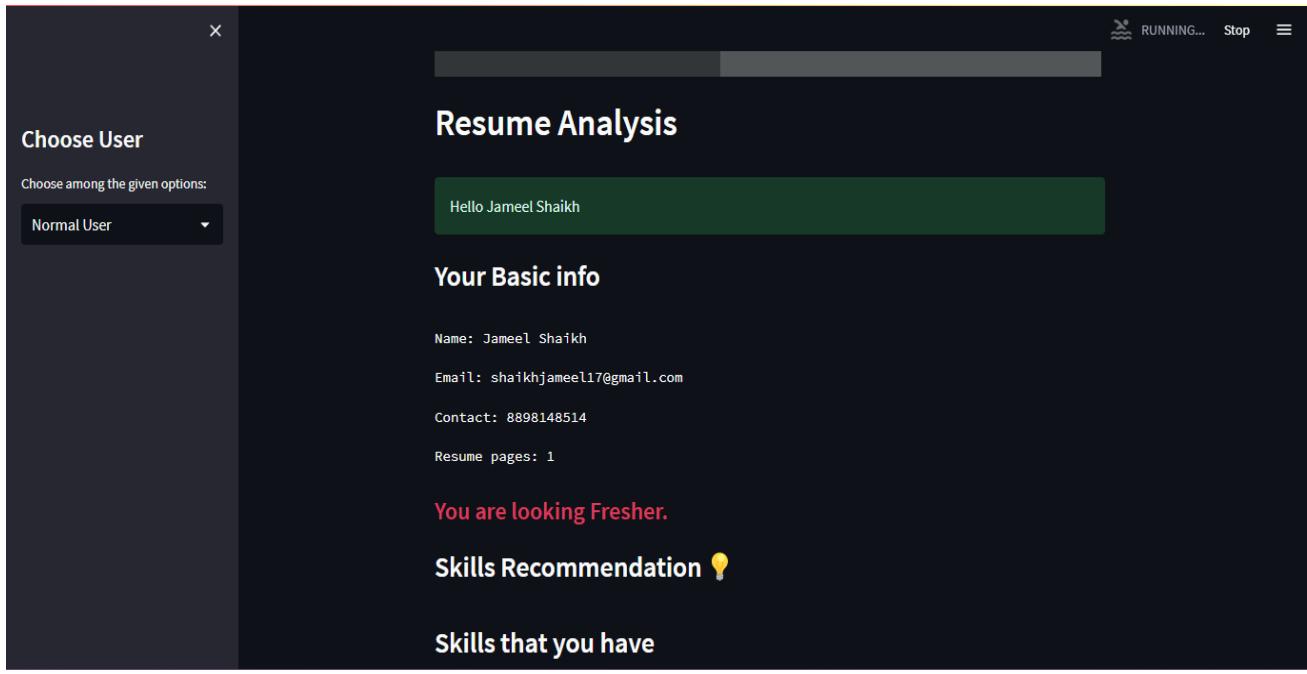


Figure 6.2: File Upload System

5.4. Screenshots:



You are looking Fresher.

Skills Recommendation 💡

Skills that you have

PHP X Javascript X Sales X Website X Html X Css X Html5 X
System X Mysql X Java X Github X Troubleshooting X Certification X
Security X Vendors X Visual X Technical X Programming X Python X
Hardware X See our skills recommendation

** Our analysis says you are looking for Web Development Jobs **

Recommended skills for you.

React X Django X Node JS X ReactJS X php X laravel X Magento X
wordpress X Javascript X Angular JS X c# X Flask X SDK X

Recommended skills generated from System

Choose User

Choose among the given options:

Normal User

Adding this skills to resume will boost 🚀 the chances of getting a Job 📦

Courses & Certificates 🎓 Recommendations

Choose Number of Course Recommendations:

4

1 10

- (1) [Full Stack Web Developer by Udacity](#)
- (2) [Become a React Developer by Udacity](#)
- (3) [Python and Django Full Stack Web Developer Bootcamp](#)
- (4) [Flask: Develop Web Applications in Python](#)

Resume Tips & Ideas 💡

[-] According to our recommendation please add your career objective, it will give your career intension to the Recruiters.

[+] According to our recommendation please add Declaration ✍ It

Choose User

Choose among the given options:

Normal User

[+] According to our recommendation please add your career objective, it will give your career intension to the Recruiters.

[+] According to our recommendation please add Declaration 📜. It will give the assurance that everything written on your resume is true and fully acknowledged by you

⌚ [+] Awesome! You have added your Hobbies 🎾

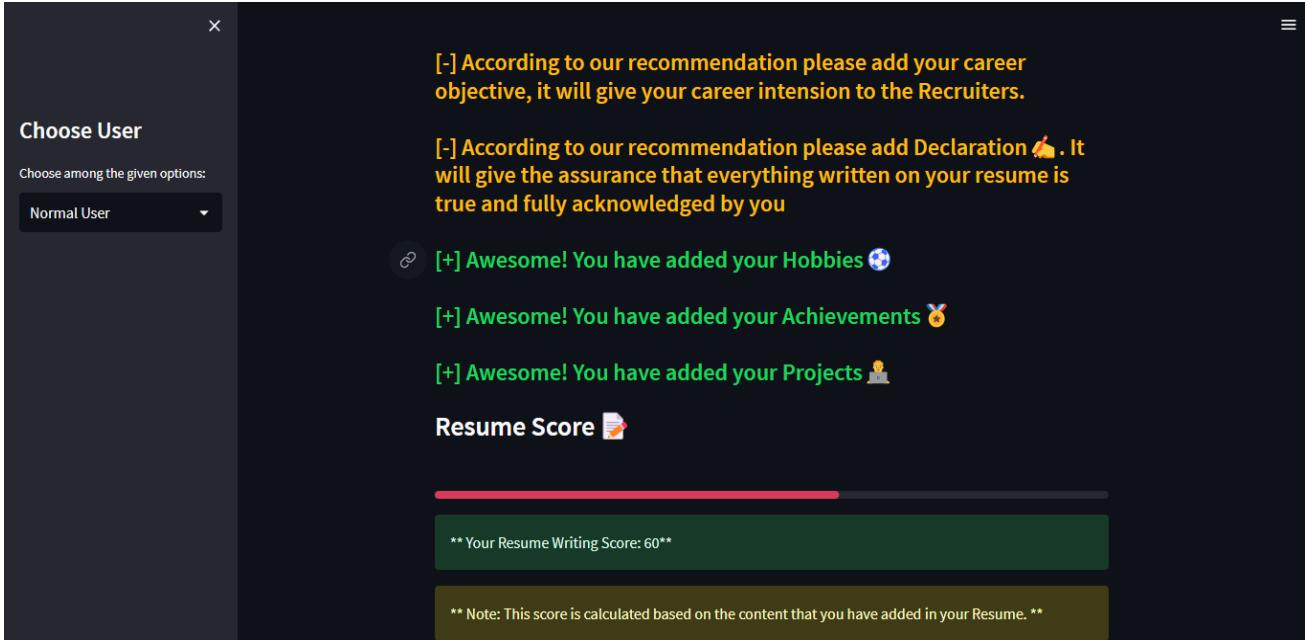
[+] Awesome! You have added your Achievements 🏆

[+] Awesome! You have added your Projects 🚀

Resume Score 📈

** Your Resume Writing Score: 60**

** Note: This score is calculated based on the content that you have added in your Resume. **



Choose User

Choose among the given options:

Normal User

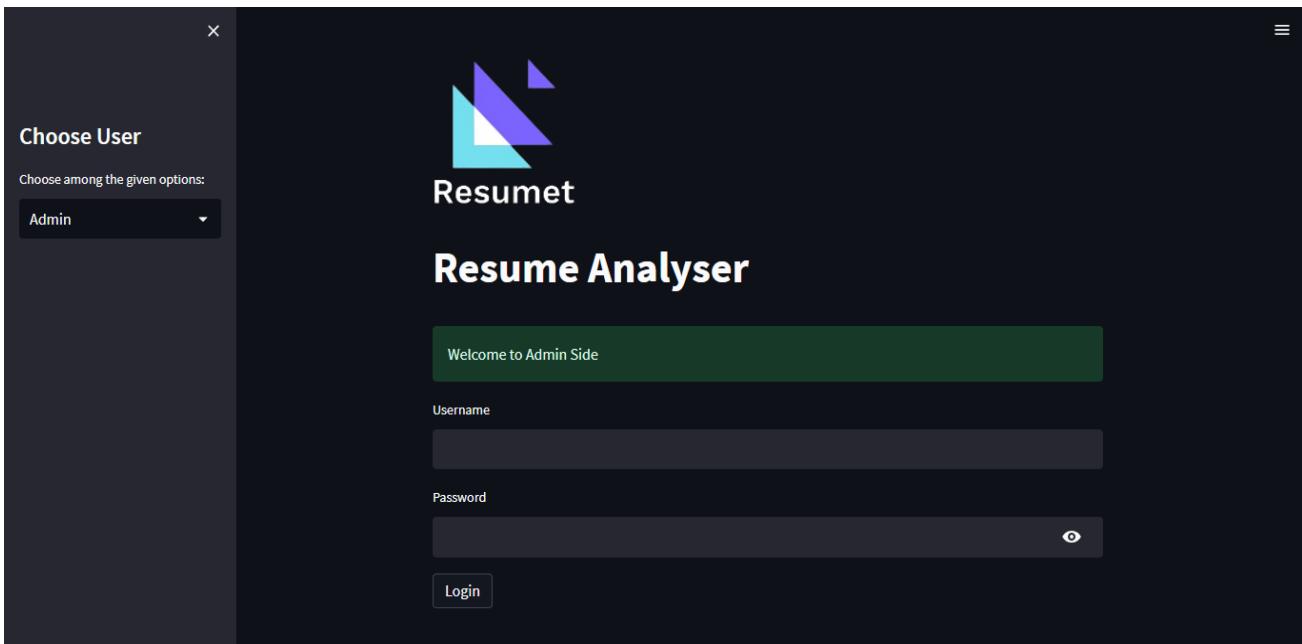
Bonus Video for Resume Writing Tips💡

How to write a Resume | Software Engineer Resume Tips For Freshers & Experienced



The video thumbnail shows a man with dark hair and a beard, looking directly at the camera. The video player interface includes a play button, a progress bar, and social sharing icons for 'Watch Later' and 'Share'.





Choose User

Choose among the given options:

Admin

Login

Welcome Jameel

User's 🧑‍💻 Data

	ID	Name	Email	Resume Score	Timestamp	Total Page	Pred
0	1	Jameel Shaikh	shaikhjameel17@gmail.com	60	2022-11-08_22:34:14	1	Web
1	2	Jameel Shaikh	shaikhjameel17@gmail.com	60	2022-11-08_22:43:45	1	Web
2	3	Jameel Shaikh	shaikhjameel17@gmail.com	60	2022-11-08_22:44:47	1	Web
3	4	Jameel Shaikh	shaikhjameel17@gmail.com	60	2022-11-12_18:29:27	1	Web

[Download Report](#)

📈 Pie-Chart for Predicted Field Recommendations

Predicted Field according to the Skills

Choose User

Choose among the given options:

Admin

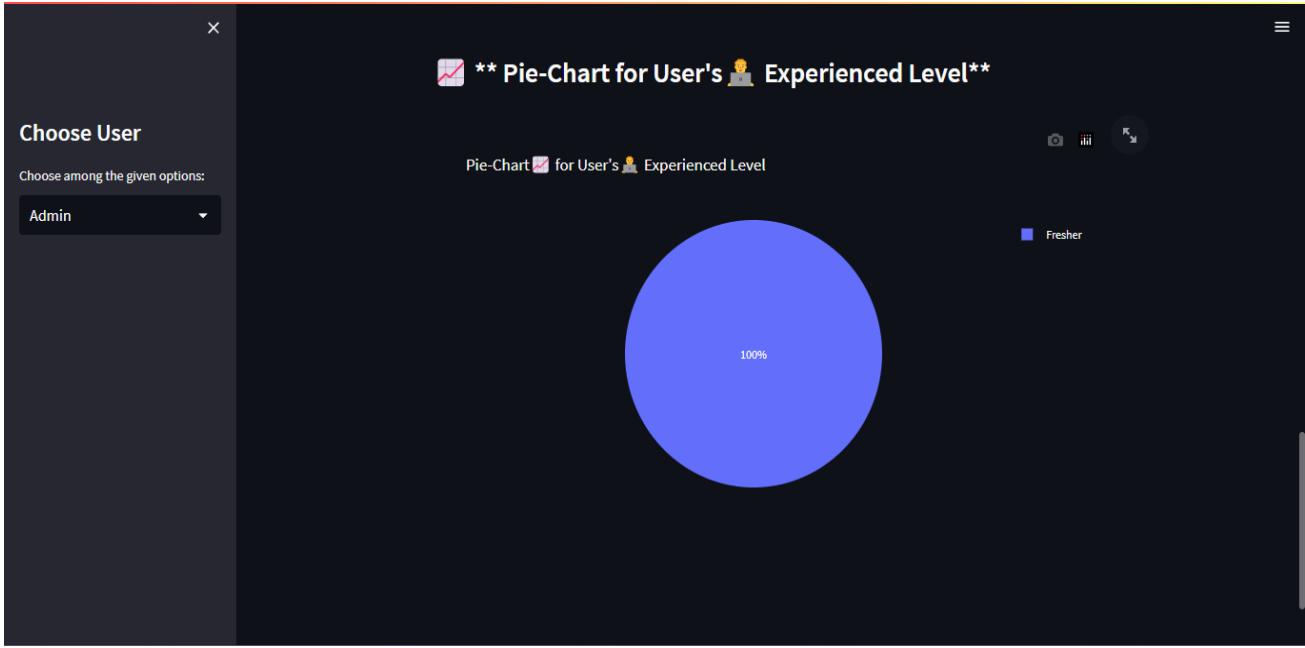
Predicted Field according to the Skills

Web Development

100%

🕒 📈 ** Pie-Chart for User's 🧑‍💻 Experienced Level**

Pie-Chart 📈 for User's 🧑‍💻 Experienced Level



5.5. Design Process

The goal of design is to get as near as possible to the final design. As a result, I built a design that featured all the pages' important content by picking the appropriate colors, font, and structure. All the visual material, including images, icons, and buttons has been created. I reduced the number of unnecessary colors that might cause users to become distracted. The design was made with Figma. My main considerations have been usability, layout, and consistency. The design has provided a general notion of what the upcoming product would look like.

5.6. Development

The final and most essential step is development. I built a functional website in development using the graphic components that were created during the wireframe. The home page is created initially, followed by all sub-pages. The static web page elements modeled during the mock-up were generated. The development languages used were Python, HTML, CSS, JavaScript and Bootstrap. The web pages are user-friendly and responsive, with plenty of visuals on screen. To make the website more engaging, various accessibility options have also been included.

5.7. Module Implementation

This part contains information about data collecting, how the code is working in the project, different libraries and modules utilized in the project, and so forth.

5.7.1 Important Necessary Libraries and Module

In today's world, everyone wants to complete the tasks as early as possible, but it all relies on the methods they use. There may be a point when we must write many lines of code; if we continue writing code on our own, this will take time. A Library is a group of interconnected modules. It provides code bundles that may be reused in a variety of programs. Modules also play a role in Python. We may define commonly used functions as modules and import them into a code wherever there is a necessity, rather than repeating the same code in multiple programs and making the code complicated. The following are some of the most important libraries and modules:

Pandas

Pandas is a Python library which is an open source. It's a tool for analyzing data.

IO

The io module is used to control file-related input and output activities. The advantage of utilizing the IO module is that we can extend the capability to allow writing to Unicode data using the classes and methods provided (journaldev, 2022).

Pdfminer3

Pdfminer is a tool that extracts text from PDF files. It focuses completely on gathering and processing text data, unlike other PDF-related tools.

Pyresparser

Pyresparser is a simple resume parser used for extracting information from resumes.

Stream Lit

Stream lit is an open-source app framework in Python language. It helps us create web apps for data science and machine learning in a short time. It is compatible with major Python libraries such as scikit-learn, Keras, PyTorch, SymPy(latex), NumPy, pandas, Matplotlib etc.

Pafy

Pafy is a Python library to download Youtube content and retrieve metadata.

Plotly

Plotly is an interactive open-source python library. It is extremely helpful tool for data visualization and understanding the data simply and easily.

PyMySQL

PyMySQL is an interface for connecting to a MySQL database server from Python. It implements the Python Database API v2.0 and contains a pure-Python MySQL client library.

Pillow

The Python Imaging Library adds image processing capabilities to your Python interpreter. This library provides extensive file format support, an efficient internal representation, and fairly powerful image processing capabilities.

Streamlit-Tags

A custom component to add tags in Stream Lit.

5.7.2 Code

```
import streamlit as st
import pandas as pd
import base64,random
import time,datetime
from pyresparser import ResumeParser
from pdfminer3.layout import LAParams, LTTextBox
from pdfminer3.pdfpage import PDFPage
from pdfminer3.pdfinterp import PDFResourceManager
from pdfminer3.pdfinterp import PDFPageInterpreter
from pdfminer3.converter import TextConverter
import io,random
from streamlit_tags import st_tags
from PIL import Image
import pymysql
from Courses import ds_course,web_course,android_course,ios_course,uiux_course,resume_videos,interview_videos
import pafy
import plotly.express as px

def fetch_yt_video(link):
    video = pafy.new(link)
    return video.title

def get_table_download_link(df,filename,text):
    """Generates a link allowing the data in a given panda dataframe to be downloaded
    in:  dataframe
    out: href string
    """
    csv = df.to_csv(index=False)
    b64 = base64.b64encode(csv.encode()).decode() # some strings <-> bytes conversions necessary here
    href = f'Download Report'
    href = f'{text}'
    return href

def pdf_reader(file):
    resource_manager = PDFResourceManager()
    fake_file_handle = io.StringIO()
    converter = TextConverter(resource_manager, fake_file_handle, laparams=LAParams())
    page_interpreter = PDFPageInterpreter(resource_manager, converter)
    with open(file, 'rb') as fh:
        for page in PDFPage.get_pages(fh,
                                      caching=True,
                                      check_extractable=True):
            page_interpreter.process_page(page)
```



```

page_title="Smart Resume Analyzer",
page_icon='./Logo/SRA_Logo.ico',
)
def run():
    st.title("Smart Resume Analyser")
    st.sidebar.markdown("# Choose User")
    activities = ["Normal User", "Admin"]
    choice = st.sidebar.selectbox("Choose among the given options:", activities)
    # link = '[©Developed by Spidy20](http://github.com/spidy20)'
    # st.sidebar.markdown(link, unsafe_allow_html=True)
    img = Image.open('./Logo/SRA_Logo.jpg')
    img = img.resize((250,250))
    st.image(img)

    # Create the DB
    db_sql = """CREATE DATABASE IF NOT EXISTS SRA;"""
    cursor.execute(db_sql)

    # Create table
    DB_table_name = 'user_data'
    table_sql = "CREATE TABLE IF NOT EXISTS " + DB_table_name + """
        (ID INT NOT NULL AUTO_INCREMENT,
        Name varchar(100) NOT NULL,
        Email_ID VARCHAR(50) NOT NULL,
        resume_score VARCHAR(8) NOT NULL,
        Timestamp VARCHAR(50) NOT NULL,
        Page_no VARCHAR(5) NOT NULL,
        Predicted_Field VARCHAR(25) NOT NULL,
        User_level VARCHAR(30) NOT NULL,
        Actual_skills VARCHAR(300) NOT NULL,
        Recommended_skills VARCHAR(300) NOT NULL,
        Recommended_courses VARCHAR(600) NOT NULL,
        PRIMARY KEY (ID));
    """
    cursor.execute(table_sql)

    if choice == 'Normal User':
        # st.markdown("<h4 style='text-align: left; color: #d73b5c;'>* Upload your resume, and get smart recommendation based on it.</h4>", unsafe_allow_html=True)
        pdf_file = st.file_uploader("Choose your Resume", type=["pdf"])
        if pdf_file is not None:
            # with st.spinner('Uploading your Resume....'):
            #     time.sleep(4)
            save_image_path = './Uploaded_Resumes/' + pdf_file.name
            with open(save_image_path, "wb") as f:

```

```

f.write(pdf_file.getbuffer())
show_pdf(save_image_path)
resume_data = ResumeParser(save_image_path).get_extracted_data()
if resume_data:
    ## Get the whole resume data
    resume_text = pdf_reader(save_image_path)

    st.header("/**Resume Analysis**")
    st.success("Hello " + resume_data['name'])
    st.subheader("/**Your Basic info**")
    try:
        st.text('Name: ' + resume_data['name'])
        st.text('Email: ' + resume_data['email'])
        st.text('Contact: ' + resume_data['mobile_number'])
        st.text('Resume pages: ' + str(resume_data['no_of_pages']))
    except:
        pass
    cand_level = ""
    if resume_data['no_of_pages'] == 1:
        cand_level = "Fresher"
        st.markdown(""

#### 


```

```

uiux_keyword = ['ux','adobe xd','figma','zeplin','balsamiq','ui','prototyping','wireframes','storyframes','adobe
photoshop','photoshop','editing','adobe illustrator','illustrator','adobe after effects','after effects','adobe premier
pro','premier pro','adobe indesign','indesign','wireframe','solid','grasp','user research','user experience']

recommended_skills = []
reco_field = ""
rec_course = ""
## Courses recommendation
for i in resume_data['skills']:
    ## Data science recommendation
    if i.lower() in ds_keyword:
        print(i.lower())
        reco_field = 'Data Science'
        st.success("*** Our analysis says you are looking for Data Science Jobs.***")
        recommended_skills = ['Data Visualization','Predictive Analysis','Statistical Modeling','Data
Mining','Clustering & Classification','Data Analytics','Quantitative Analysis','Web Scraping','ML
Algorithms','Keras','Pytorch','Probability','Scikit-learn','Tensorflow','Flask','Streamlit']
        recommended_keywords = st_tags(label='### Recommended skills for you.',

text='Recommended skills generated from System',value=recommended_skills,key = '2')
        st.markdown("<h4 style='text-align: left; color: #1ed760;'>Adding this skills to resume will boost 💪
the chances of getting a Job </h4>","",unsafe_allow_html=True)
        rec_course = course_recommender(ds_course)
        break

## Web development recommendation
elif i.lower() in web_keyword:
    print(i.lower())
    reco_field = 'Web Development'
    st.success("*** Our analysis says you are looking for Web Development Jobs ***")
    recommended_skills = ['React','Django','Node JS','React
JS','php','laravel','Magento','wordpress','Javascript','Angular JS','c#','Flask','SDK']
    recommended_keywords = st_tags(label='### Recommended skills for you.',

text='Recommended skills generated from System',value=recommended_skills,key = '3')
    st.markdown("<h4 style='text-align: left; color: #1ed760;'>Adding this skills to resume will boost 💪
the chances of getting a Job </h4>","",unsafe_allow_html=True)
    rec_course = course_recommender(web_course)
    break

## Android App Development
elif i.lower() in android_keyword:
    print(i.lower())
    reco_field = 'Android Development'
    st.success("*** Our analysis says you are looking for Android App Development Jobs ***")
    recommended_skills = ['Android','Android
development','Flutter','Kotlin','XML','Java','Kivy','GIT','SDK','SQLite']

```

```

recommended_keywords = st_tags(label='### Recommended skills for you.',  

text='Recommended skills generated from System',value=recommended_skills,key = '4')  

st.markdown("<h4 style='text-align: left; color: #1ed760;'>Adding this skills to resume will boost ↗  

the chances of getting a Job ☺</h4>","",unsafe_allow_html=True)  

rec_course = course_recommender(android_course)  

break

## IOS App Development  

elif i.lower() in ios_keyword:  

    print(i.lower())  

    reco_field = 'IOS Development'  

    st.success("*** Our analysis says you are looking for IOS App Development Jobs **")  

    recommended_skills = ['IOS','IOS Development','Swift','Cocoa','Cocoa Touch','Xcode','Objective-C','SQLite','Plist','StoreKit','UI-Kit','AV Foundation','Auto-Layout']  

    recommended_keywords = st_tags(label='### Recommended skills for you.',  

text='Recommended skills generated from System',value=recommended_skills,key = '5')  

    st.markdown("<h4 style='text-align: left; color: #1ed760;'>Adding this skills to resume will boost ↗  

the chances of getting a Job ☺</h4>","",unsafe_allow_html=True)  

rec_course = course_recommender(ios_course)  

break

## UI-UX Recommendation  

elif i.lower() in uiux_keyword:  

    print(i.lower())  

    reco_field = 'UI-UX Development'  

    st.success("*** Our analysis says you are looking for UI-UX Development Jobs **")  

    recommended_skills = ['UI','User Experience','Adobe  

XD','Figma','Zeplin','Balsamiq','Prototyping','Wireframes','Storyframes','Adobe Photoshop','Editing','Illustrator','After  

Effects','Premier Pro','InDesign','Wireframe','Solid','Grasp','User Research']  

    recommended_keywords = st_tags(label='### Recommended skills for you.',  

text='Recommended skills generated from System',value=recommended_skills,key = '6')  

    st.markdown("<h4 style='text-align: left; color: #1ed760;'>Adding this skills to resume will boost ↗  

the chances of getting a Job ☺</h4>","",unsafe_allow_html=True)  

rec_course = course_recommender(uiux_course)  

break

#  

## Insert into table  

ts = time.time()  

cur_date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')  

cur_time = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')  

timestamp = str(cur_date+'_'+cur_time)

### Resume writing recommendation  

st.subheader("**Resume Tips & Ideas ☀ **")

```

```

resume_score = 0
if 'Objective' in resume_text:
    resume_score = resume_score+20
    st.markdown("<h4 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added Objective</h4>",unsafe_allow_html=True)
else:
    st.markdown("<h4 style='text-align: left; color: #fabc10;'>[-] According to our recommendation please add your career objective, it will give your career intension to the Recruiters.</h4>",unsafe_allow_html=True)

if 'Declaration' in resume_text:
    resume_score = resume_score + 20
    st.markdown("<h4 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added Declaration</h4>",unsafe_allow_html=True)
else:
    st.markdown("<h4 style='text-align: left; color: #fabc10;'>[-] According to our recommendation please add Declaration. It will give the assurance that everything written on your resume is true and fully acknowledged by you</h4>",unsafe_allow_html=True)

if 'Hobbies' or 'Interests'in resume_text:
    resume_score = resume_score + 20
    st.markdown("<h4 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added your Hobbies 🎾</h4>",unsafe_allow_html=True)
else:
    st.markdown("<h4 style='text-align: left; color: #fabc10;'>[-] According to our recommendation please add Hobbies 🎾. It will show your personality to the Recruiters and give the assurance that you are fit for this role or not.</h4>",unsafe_allow_html=True)

if 'Achievements' in resume_text:
    resume_score = resume_score + 20
    st.markdown("<h4 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added your Achievements 💯 </h4>",unsafe_allow_html=True)
else:
    st.markdown("<h4 style='text-align: left; color: #fabc10;'>[-] According to our recommendation please add Achievements 💯. It will show that you are capable for the required position.</h4>",unsafe_allow_html=True)

if 'Projects' in resume_text:
    resume_score = resume_score + 20
    st.markdown("<h4 style='text-align: left; color: #1ed760;'>[+] Awesome! You have added your Projects 🏢 </h4>",unsafe_allow_html=True)
else:
    st.markdown("<h4 style='text-align: left; color: #fabc10;'>[-] According to our recommendation please add Projects 🏢. It will show that you have done work related the required position or not.</h4>",unsafe_allow_html=True)

st.subheader("'''Resume Score 💬'''")

```

```

st.markdown(
    """
<style>
    .stProgress > div > div > div > div {
        background-color: #d73b5c;
    }
</style>"",
    unsafe_allow_html=True,
)
my_bar = st.progress(0)
score = 0
for percent_complete in range(resume_score):
    score +=1
    time.sleep(0.1)
    my_bar.progress(percent_complete + 1)
st.success(** Your Resume Writing Score: ' + str(score)+ '**')
st.warning("** Note: This score is calculated based on the content that you have added in your Resume. **")
st.balloons()

insert_data(resume_data['name'], resume_data['email'], str(resume_score), timestamp,
            str(resume_data['no_of_pages']), reco_field, cand_level, str(resume_data['skills']),
            str(recommended_skills), str(rec_course))

## Resume writing video
st.header("**Bonus Video for Resume Writing Tips 🎬 **")
resume_vid = random.choice(resume_videos)
res_vid_title = fetch_yt_video(resume_vid)
st.subheader("☒ **"+res_vid_title+"**")
st.video(resume_vid)

## Interview Preparation Video
st.header("**Bonus Video for Interview 🎤 Tips 🎬 **")
interview_vid = random.choice(interview_videos)
int_vid_title = fetch_yt_video(interview_vid)
st.subheader("☒ ** "+ int_vid_title + "**")
st.video(interview_vid)

connection.commit()
else:
    st.error('Something went wrong..')
else:
    ## Admin Side
    st.success('Welcome to Admin Side')
    # st.sidebar.subheader(**ID / Password Required!**)

```

```

ad_user = st.text_input("Username")
ad_password = st.text_input("Password", type='password')
if st.button('Login'):
    if ad_user == 'machine_learning_hub' and ad_password == 'mlhub123':
        st.success("Welcome Kushal")
    # Display Data
    cursor.execute("SELECT*FROM user_data")
    data = cursor.fetchall()
    st.header("User's Data")
    df = pd.DataFrame(data, columns=['ID', 'Name', 'Email', 'Resume Score', 'Timestamp', 'Total Page',
                                     'Predicted Field', 'User Level', 'Actual Skills', 'Recommended Skills',
                                     'Recommended Course'])
    st.dataframe(df)
    st.markdown(get_table_download_link(df,'User_Data.csv','Download Report'), unsafe_allow_html=True)
## Admin Side Data
query = 'select * from user_data;'
plot_data = pd.read_sql(query, connection)

## Pie chart for predicted field recommendations
labels = plot_data.Predicted_Field.unique()
print(labels)
values = plot_data.Predicted_Field.value_counts()
print(values)
st.subheader("Pie-Chart for Predicted Field Recommendations")
fig = px.pie(df, values=values, names=labels, title='Predicted Field according to the Skills')
st.plotly_chart(fig)

### Pie chart for User's Experienced Level
labels = plot_data.User_level.unique()
values = plot_data.User_level.value_counts()
st.subheader("Pie-Chart for User's Experienced Level")
fig = px.pie(df, values=values, names=labels, title="Pie-Chart for User's Experienced Level")
st.plotly_chart(fig)

else:
    st.error("Wrong ID & Password Provided")
run()

```

6. Conclusion

A normal resume is a compilation of information about a person's work experience, academic background, qualifications, and personal details. These elements might be present in a variety of ways or not at all. It is difficult to keep up with the jargon used in resumes. A resume is made up of corporate names, institutions, degrees, and other information that can be written in a variety of ways. It will take time to review all the resume by an individual.

Machine works faster than human and their accuracy to do any task was also good. Therefore, I have made a system which includes machine learning that extract the important information from resumes within a minute or less than a minute. The hiring individual can use this system for hiring any individual.

7. Critical Evaluation

7.1. Final Report

While working on my report, I experienced several issues. The primary issue was that I got tiny bit confused throughout the testing part. The diagrams and grouping of some topics into subtopics were the next difficult phase. However, after discussing with the teacher, I was able to resolve this issue. Aside from the flaws, the positive aspect of the report is that it is based on facts, researched information, and legitimate proofs.

7.2. Finding and process

There was some uncertainty when it came to select the right libraries and modules for data extraction. After conducting research, I was able to select a library. Following that, obtaining the appropriate dataset was difficult. I looked for a number of resumes dataset but couldn't find one in the correct format. After that, I went online and found several templates and started using them for training data.

7.3. System

While developing the system, there were a number of technical difficulties. Following the development of the module, the next step is to integrate it into the system. I was having trouble

integrating the module while working on the flask. I tried everything but couldn't get the component to work. As a result, I switched to Django as my framework. I was able to integrate after reading through several python and Django documentation. The integration part took up more time than expected.

Despite the time constraints, the majority of the features proposed in the proposal were implemented. However, the Tika parser library, which I specified in the proposal for the information extraction section, has not been implemented. It is used to detect the kind of document. As I began working on the project, I found that Spacy and Nltk were more suitable to my requirements, so I chose them over Tika.

7.4. Future Work

As stated in the paper, the project has a broad reach in the current context. The proposal's majority of proposed features have been implemented. So, if I continue working on this project, I intend to create a database for the system where the admin may keep the extracted data. Further, future study will include a more in-depth examination of certain techniques, further research on other libraries, and new approaches to explore different methods.

7.5. Self-reflection

After working on a project, everyone learns something new. After completing this project, I learned several new things about information extraction. I learnt how to deal with various NLP libraries, how to implement the module, and so on. Aside than that, I learnt about Django by watching videos and reading documentation. I also learned how to integrate the system from this portion.

8. References

- C, B. P., 2020. *towardsai*. [Online]
Available at: <https://towardsai.net/p/nlp/natural-language-processing-concepts-andworkflow-48083d2e3ce7> [Accessed 2022].
- Agrawal , R., 2021. *analyticsvidhya*. [Online]
Available at: <https://www.analyticsvidhya.com/blog/2021/06/must-known-techniques-for-text-preprocessing-in-nlp/#:~:text=Text%20preprocessing%20is%20a%20method,text%20in%20a%20different%20case> [Accessed 2022].
- Bhatia, V., Rawat, P., Kumar, A. & Shah, R. R., 2019. End-to-End Resume Parsing and Finding Candidates for a Job Description using BERT. *arxiv*.
- chavan, J., 2020. *medium*. [Online]
Available at: <https://medium.com/@jeevanchavan143/nlp-tokenization-stemming-lemmatization-bag-of-words-tf-idf-pos-7650f83c60be> [Accessed 2022].
- Chen, J., Gao, . L. & Tang, Z., 2016. Information Extraction from Resume Documents in PDF. *researchgate*.
- Kopparapu, S. . K., 2015. Automatic Extraction of Usable Information from Unstructured Resumes to Aid Search. *ieeexplore*.
- Nguyen, V. V., Pham, V. . L. & Vu, N. . S., 2018. Study of Information Extraction in Resume. *semanticscholar*.
- D., 2021. *analyticsvidhya*. [Online]
Available at: <https://www.analyticsvidhya.com/blog/2021/06/text-preprocessing-in-nlp-with-python-codes/> [Accessed 2022].
- geeksforgeeks, 2018. *geeksforgeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/> [Accessed 2022].
- geeksforgeeks, 2018. *geeksforgeeks*. [Online]
Available at: <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/#:~:text=An%20activity%20diagram%20is%20a,the%20activity%20is%20being%20executed> [Accessed 2022].
- ibm, 2021. *ibm*. [Online]

Available at: <https://www.ibm.com/docs/en/rational-soft-arch/9.6.1?topic=diagramsuse-case>

[Accessed 2022].

- journaldev, 2022. Python IO Module. *journaldev*.
- Kurama, V., 2021. *nanonets*. [Online]
Available at: <https://nanonets.com/blog/information-extraction/> [Accessed 2022].
- teamwork, 2021. *teamwork*. [Online]
Available at: <https://www.teamwork.com/project-management-guide/projectmanagement-methodologies/> [Accessed 2022].
- VUKADIN, D., KURDIJA, A. . S., DELAČ, G. & ŠILIĆ, M., 2021. Information Extraction From Free-Form. *ieeexplore*.
- Wang, X. & Zu, S., 2019. RESUME INFORMATION EXTRACTION WITH ANOVEL TEXT BLOCK SEGMENTATION ALGORITHM. *researchgate*, Volume 8.
- Wolff, R., 2020. *monkeylearn*. [Online]
Available at: <https://monkeylearn.com/blog/nlp-benefits/> [Accessed 2022].