A Project Report on

Smart Resume Analyzer

Submitted in partial fulfillment for award of

Bachelor of Technology

Degree in

Computer Science and Engineering

By

M. Swathi (Y20ACS493) **M. Subhash Reddy (Y20ACS496)**

L. Sekhar (Y20ACS490) **K. Likitha (Y20ACS474)**

Under the guidance of

Prof. K. Kishan Chand.

Department of Computer Science and Engineering

Bapatla Engineering College

(Autonomous)

(Affiliated to Acharya Nagarjuna University)

BAPATLA – 522 102, Andhra Pradesh, INDIA

2023-2024

Department of
Computer Science and Engineering

CERTIFICATE

This is to certify that the project report entitled **Smart Resume Analyzer** that is being submitted by M. Swathi (Y20ACS493), M. Subhash Reddy (Y20ACS496), L. Sekhar (Y20ACS490) and K. Likitha (Y20ACS474) in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science & Engineering to the Acharya Nagarjuna University is a record of bonafide work carried out by them under our guidance and supervision.

Date:

Signature of the Guide

K. Kishan Chand

Associate Professor

Signature of the HOD

Dr. M. Rajesh Babu.

Prof. & Head

DECLARATION

We declare that this project work is composed by ourselves, that the work contained herein is our own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

M. Swathi (Y20ACS493)

M. Subhash Reddy(Y20ACS496)

L. Sekhar (Y20ACS490)

K. Likitha (Y20ACS474)

Acknowledgement

We sincerely thank the following distinguished personalities who have given their advice and support for successful completion of the work.

We are deeply indebted to our most respected guide **[K. Kishan Chand, Associate Professor],** Department of CSE, for his/her valuable and inspiring guidance, comments, suggestions and encouragement.

We extend our sincere thanks to **Dr. K. Rajesh Babu**, Prof. & Head of the Dept. for extending his cooperation and providing the required resources.

We would like to thank our beloved Principal [**Dr. Sk. Nazeer]** for providing the online resources and other facilities to carry out this work.

We would like to express our sincere thanks to our project coordinator **Dr. K. Manideep,** Prof. Dept. of CSE for his helpful suggestions in presenting this document.

We extend our sincere thanks to all other teaching faculty and non-teaching staff of the department, who helped directly or indirectly for their cooperation and encouragement.

M. Swathi (Y20ACS493)

M. Subhash Reddy (Y20ACS496)

L. Sekhar (Y20ACS490)

K. Likitha (Y20ACS47

Table of Contents

List of Tables

List of Figures

Abstract

The Resume Analyzer is a web application based on Streamlit that can analyze a PDF-format resume uploaded by the user. It extracts information and returns a score based on the presence of key elements such as a declaration, hobbies, and qualifications. The analyzer reduces the need for manual resume checking and boosts the user's confidence. It works using natural language processing and text mining to deduce information from the resume. The project called "SRA: Resume Analyzer" is aimed at analyzing any person's resume and providing recommendations to the user based on the analysis. This is a user-friendly and intelligent application where users only need to upload their resumes, and the system will automatically analyze them and provide recommendations.

Keywords:

Natural language Processing, Streamlit, Text Mining, Pdfminer, Pyresparser.

1 Introduction

In today's world, many people possess skills, yet struggle to find highly-paid jobs. One of the main reasons for this is the lack of quality projection of their resumes during interviews. To help fresher candidates present the perfect resume that accurately reflects their skills, our resume analyzer uses NLP to analyze and provide feedback. It's crucial to make a good impression on recruiters with your resume as it's often your only chance to receive an interview invitation. Even personal consultations with experienced HR managers do not guarantee that your resume will be accepted by ATS. By running your resume through our review tool, you can identify common problems, learn about formatting, and get helpful tips on how to stand out to recruiters. Our Smart Resume Analyzer System is a text-mining application that analyzes resumes using keyword-matching algorithms. It maps keywords from a personalized dictionary against the terms in the resume, extracts necessary data, and stores it in a database. The system sorts the information based on experience, education and other criteria. In today's competitive job market, having the right skills is not always enough to secure a highly-paid job. One of the biggest challenges people face is crafting a compelling resume that accurately reflects their abilities and experience. Our Resume Analyzer tool is designed to help you do just that. Using advanced NLP technology, it analyzes your resume and provides valuable feedback to help you make a great impression on recruiters.

Your resume is often the first thing recruiters see, and it's essential to make it stand out. Even if you've had consultations with HR professionals, it's not always easy to know what recruiters are looking for. Our tool takes the guesswork out of the equation

by analyzing your resume's formatting, identifying common issues, and providing tips to help you get noticed.

Our Smart Resume Analyzer System is a powerful text-mining application that uses keyword-matching algorithms to extract relevant data from your resume and store it in a database. By mapping keywords against a personalized dictionary, the system can sort information based on factors such as experience, education, and age. This allows us to schedule interviews more efficiently and notify you of your interview via automated emails.

In short, our Resume Analyzer tool is an invaluable resource for anyone looking to land their dream job. Whether you're a recent graduate or a seasoned professional, our tool can help you craft a winning resume that showcases your skills and experience in the best possible light.

2 Problem Definition

In the present competitive job market, job seekers and employers encounter challenges in efficiently matching skills and qualifications. Job seekers often face difficulties in highlighting their relevant experiences and skills in their resumes, while employers receive an overwhelming number of resumes for each job opening, making it time-consuming to identify the most suitable candidates. Additionally, the lack of personalized feedback makes it challenging for job seekers to understand how to improve their application materials for better success rates. They often receive minimal or generic feedback on their resumes, which further hampers their chances of securing a job. Therefore, in today's competitive job market, both job seekers and employers require assistance in effectively matching skills and qualifications. In the current job market, finding the right match between job seekers and employers can be a challenge. Job seekers often struggle to showcase their skills and experiences effectively in their resumes, while employers receive a large number of resumes for each job opening, making it difficult to identify the most suitable candidates. Furthermore, job seekers often receive generic or minimal feedback on their application materials, making it challenging for them to improve their success rates.

3 Literature Survey

The employment process is being revolutionized in the modern world by the emergence of technology like the Internet. More than fifty thousand Internet employment portals invite candidates to upload their resumes. The likelihood of your application being chosen among thousands of others is quite low. Using this clever resume analyzer could significantly increase your chances of being added to the candidate list. It provides advice on resume writing techniques.

The CV RDF Ontology was created by John G. Breslin and Uldis Bojars. It is a model of the CV that utilizes the RDF data model. The Resume RDF describes resume-related information using a wide range of classes and attributes. To better explain the material, Uldis Bojars supplemented the FOAF with additional details. In 2002 and 2003, Turney and Littmann considered the semantic orientation or evaluation features of words from a corpus of 100 billion words. They provided a schema for guessing and also considered the semantic relevance of other words identified as a paradigm.

Using the concept of linked data, Ujjal Marjit et al. [1] proposed an additional method for recovering CV information. This makes it possible for the Internet to find and exchange many kinds of information.

Zhi Xiang Jing et al. created an online Chinese resume parser [2] that used a rule-based statistical algorithm to extract data from the CV. By working on block analysis of CV documents based on pattern matching and multi-level information identification, Zhang Chuang et al. [3] created the largest CV analysis system.

The following system was designed by Elik et al. [4] to translate CV to the ontological framework. Turkish and Turkish analysis English CV simplified in a model.

Di Wu and associates [5] Resultant Improved resume information extraction with WordNet Calculation of Similarity and the ontology concept.

Top resume analyzed using techniques such as natural language processing. With the help of natural language processing, only the text data is extracted from the CV and the strength of the applicant profile is displayed as a percentage. Additional attributes such as B. Percentage of candidate skills as a result of a resume review, depending on the candidate's education, qualifications, courses, and work experience. However, there is no provision on the website that it applies only to job seekers. Posting a particular job may give recruiters within your organization the opportunity to provide a ranking of candidates that match your immediate job-related skills. Therefore, there are many other such web applications, most of which provide similar functionality.

The Cluster-based Ranking Index (CBR), which evaluates resumes to identify the top candidates, was introduced by Mayuri Verma as a way to enhance the hiring process. [6] The development of a recommender system received very little attention in the past. Nonetheless, a great deal of study on recommender systems has been identified. The first recommendation system was created by Resnick et al. [7].

Otaibi et al. examined the use of recruitment services, outlined the steps that should be taken in each organization's hiring process, discussed the advantages of using electronic recruitment portals, what factors should be considered when selecting candidates, and a host of other crucial hiring strategies. A text mining study employing Natural Language Processing (NLP) and clustering was presented by R. Janani et al. At the extraction point, text mining methods were used to boost efficiency. A study titled "Toward an Ontology-Based Information Extraction System that Matches Work with CV" was published by D. Cerrick and colleagues. In this case, the Ontology Knowledge Base transforms a resume written in plain text into an Ontology form (OKB). The

completion rate is determined by this system based on factors including education and work experience. Marinovsky and others to use an Expectation-Maximization (EM) algorithm to draft an employment proposal that takes into account the job description provided by the company as well as the candidate's resume. Golecetal suggests using the fuzzy approach to assess a candidate's job description relevancy. The rapidly expanding discipline of text mining, which is described as statistical machine learning how to interpret unstructured raw materials, is also explained by text analysis.

For additional classification, classification, adequate training, and training, you can convert data into structured data. Excellent information for functionality [8]. technology. Text mining, also known as Natural Language Processing (NLP), involves pattern identification, information extraction, data mining, and parsing. Even if there are several other websites for resume analysis, our website is unique in its way. In our we have two interfaces one is the admin interface and the other is the user interface. In the admin interface, you view the candidates who have used our website in an excel. Recruiters can put the user's resume on the candidate list based on their resume score and the skills required for a particular position. In this way, it helps the recruiters. On the other hand, you have a user interface where you can upload your resume and get the resume score and the recommendations required to boost your resume score.

4 System Analysis

System analysis is a critical phase in the software development lifecycle where the requirements, constraints, and objectives of a project are carefully examined and documented.

4.1 Objective

The purpose of this project is to develop a smart technology for the corporate hiring world. With thousands of people facing unemployment and struggling to find work, we aim to create a system that can provide personalized recommendations based on the user's resume. These recommendations will include the best job industries, necessary tools and technologies, relevant courses, and effective resume writing techniques. This will help users improve their resumes and increase their chances of securing a job in a reputable company.

4.2 Existing System

The existing system uses Naïve Bayes algorithm. This method uses traditional machine learning and has lower accuracy rates. It is less effective and inaccurate data. The potential of people may be lost due to this system. The recruiters cannot differentiate between the similarity of the resumes of the candidates as they might have taken a copy of someone else’s resume. There are other approaches being tried, such KNN, however they are ineffective for huge datasets. Another popular approach is text parsing; however it can only be applied to resumes with structure. If the outcome is positive, the resume matched the job description, then the test of hypothesis gives a statistical representation by comparing the variable data to the specified data. Manually

checking the resumes can be time-consuming for the recruiters. Although semi-supervised learning based on machine learning can be utilised, it can only forecast the quality of the resume based on the job description.

4.3 Limitations of Existing Methods

a. The drawbacks in all the existing papers used ML Techniques & some algorithms after screening or parsing the data that is unstructured or structured data.
b. It accepts or rejects the applications it only matches to the conditions set by the recruiters it doesn’t give any type of recommendation.
c. These are the few drawbacks of the existing papers.

4.4 Proposed System

A significant number of Python modules are used in Resume Analyzer's operation, which facilitates the application's management and functionality. Streamlit, Resume Parser, pandas, pdfminer3, matplotlib, and Pysql are a few of the noteworthy modules. The roles of the various components are as follows: Streamlit is more structured and simplicity-focused, and it is used to quickly construct online apps for data science and machine learning. Thus, the smart resume analyzer uses Streamlit.

The main purpose of building this project is to create a smart technology for the corporate hiring world. Nowadays thousands of people are unemployed and facing problems in finding jobs. We aim to provide a smart system that can give the perfect recommendation based on the resume of the user, It will recommend the Job working industry, tools and Technology, courses, and resume writing techniques. So, with this

help, users can create a better resume which will increase their chances of getting a job in a good company. The purpose of this project is to develop a smart technology for the corporate hiring world. With thousands of people facing unemployment and struggling to find work, we aim to create a system that can provide personalized recommendations based on the user's resume. These recommendations will include the best job industries, necessary tools and technologies, relevant courses, and effective resume-writing techniques. This will help users improve their resumes and increase their chances of securing a job in a reputable company.

Web applications continue to evolve at an incredible rate, and the architecture around web applications is becoming more and more complex. Most web applications rely on client-server architecture, where the client gives information, and the server stores and retrieves the information. Most web applications available on the Internet are written in programming languages such as HTML, and CSS for designing and animations, and Javascript is used to create the front-end interface (client-side program). For scripting web applications, server-side programming is written using programming languages like Python, Java, PHP, Ruby, etc. Python and Java are commonly used languages for server-side programming. A resume analyzer is a Streamlit-based web application that analyzes an uploaded resume in the format of a pdf and extracts information. This analysis is done with the help of Python libraries that include pdfminer3, nltk, spacy, pandas, NumPy, etc. The web application has two sections, mainly the normal user and admin panel. The user can upload his/her resume in the user section with a limit of PDF files up to 200 megabytes. Upon upload, the site user will be able to see his/her resume and the analysis of the resume which includes the user's current skills, recommended skills, resume score of the user, and the level of the user as per the analysis.

This analysis helps the user to check for improvements in his/her resume which could help him to get better opportunities. The resume data which is uploaded now gets stored in the local mysql database which can be accessed by the admin only through username and password. The admin section which unlocks through a password has a visual analysis of the total resume; this analysis is represented in the form of pie charts. Using the plot library of Python. The data of users can also be downloaded in the form of an Excel (.csv format file) which provides the scope for further personalized analysis for specific purposes. The application can be best used by recruiters to get resumes from the applicants and they can sort the applicants very easily using our application. For example, if the recruiter receives hundreds of applications then manually checking them could be a very cumbersome task through a resume analyzer the process could be done smoothly.

To identify most qualified candidates for a particular position for any job opening. Useful data from the resume, e.g., education, skills, achievements, experience and so on, should be automatically generated by the tool. To shorten recruitment time and reduce bias during the selection process, with a view to ranking candidates based on various aspects of their resume.

Features of Proposed System

In the current system, we have a resume uploader that allows users to upload their resumes. After the resume is uploaded, the system stores it and converts all resume pages into images. The next step is to extract text from these images using natural language processing (NLP). Based on the extracted text, the system provides recommendations to the user regarding skills, courses, and resume writing. Users also

have the option to remove their resumes from the system, although their data will be stored for future reference and training purposes.

Feasibility Study

The feasibility analysis begins by defining the project goals and generating possible solutions to provide an indication of the new system's potential. This phase requires creativity and imagination to explore new ways of doing things and generate ideas.

Technical Feasibility

The application is developed as a web application and requires specific tools and technologies. The main technology used is natural language processing (NLP), implemented in Python programming language. Adobe XD is used for design purposes, and Star UML is used for creating diagrams. The PyCharm IDE is utilized for development.

Financial Feasibility

The application is freely available for all users, without any charges for usage. There are no monetary services associated with the application, allowing every user to access it freely.

Operational Feasibility

Operational feasibility measures how well the proposed system solves problems and takes advantage of opportunities identified during scope definition. It ensures that the system satisfies the requirements identified in the requirements analysis phase. In our application, all operations, such as PDF processing and system recommendations,

are functioning properly. The admin can access all user data, and users can remove their resumes from the system.

4.5 Hardware Requirements

The following are the system requirements to develop Smart Resume Analyzer.

a. Processor: Intel Core i5
b. Hard Disk: Minimum 100GB
c. RAM: Minimum 8GB

4.6 Software Requirements

The following are the softwares used in the development of the app.

a. Operating System: Windows or Linux

Python Programming Language

Python is advanced, higher level and easy to learn programming language, We have chosen Python to develop Desktop GUI, We will create a desktop application using Python. In python everything is related to the Image processing is very easy to implement. The documentation and community of the python is very big, so we will get the help in development from the community.

PyCharm IDE

PyCharm is professional IDE which is developed by JetBrains. The features which are offered by PyCharm Community version is Intelligent Python editor, Code debugger, Code inspection, VCS support and many more related to the User Interface.

NLTK

NLTK, short for Natural Language Toolkit, is a comprehensive library for natural language processing (NLP) tasks in Python. It provides a wide range of tools and resources for processing and analyzing human language data, making it invaluable for tasks such as text classification, tokenization, stemming, tagging, parsing, and sentiment analysis. NLTK includes a diverse collection of corpora, lexical resources, and pre-trained models for various languages and domains, enabling developers to perform NLP tasks with ease. One of NLTK's key strengths is its simplicity and ease of use, making it accessible to both novice and experienced NLP practitioners. Its modular design allows users to mix and match components to suit their specific needs, whether it's basic text processing or advanced linguistic analysis. Additionally, NLTK is actively maintained and has a vibrant community, ensuring ongoing support, updates, and contributions from developers worldwide. Whether you're a researcher, educator, or industry professional, NLTK provides a powerful and flexible platform for exploring, experimenting, and implementing NLP solutions in Python.

XAMP

It is open source control panel which is easy to use and maintain the apache, database, and PhpMyAdmin. It is very easy to operate. We can operate apache, MySQL easily from it. It is easily available for windows and other platforms.

GitHub

It is free and open source backup platform for your running project. It is maintaining by GIT. It’s very easy to upload your work on free. There is no storage

limitation. You can do development in group and separately also. It is very easy to maintain the Information Technology work with other teammates also.

Visual

Visual Studio Code, commonly referred to as VS Code, is a renowned and widely used source code editor developed by Microsoft. It stands out for its versatility, lightweight design, and rich ecosystem of extensions. Available across Windows, macOS, and Linux, VS Code's cross-platform compatibility has made it a favored choice among developers from various backgrounds. Notably, its lightweight nature ensures swift startup times and efficient use of system resources, even on less powerful machines.

What sets VS Code apart is its extensibility. It boasts a vast collection of extensions, enabling developers to tailor the editor to their specific needs. These extensions span different programming languages, frameworks, and tools, enhancing functionality and personalizing the coding environment. Despite being a code editor, VS Code offers integrated development environment (IDE) features, including code navigation, debugging, version control integration, and an integrated terminal.

Intelligent code editing features such as IntelliSense (code completion), code refactoring, and real-time error checking assist developers in writing clean, error-free code. The built-in Git support simplifies version control management and fosters collaboration. It also provides debugging capabilities for multiple programming

languages, allowing users to set breakpoints, inspect variables, and debug their code efficiently.

VS Code's integrated terminal is a time-saving feature that enables running commands and scripts without the need to switch to a separate terminal window. Its multi-language support, encompassing syntax highlighting, code formatting, and linting, caters to a diverse range of developers. Furthermore, it offers extensive customization options, including a variety of themes and appearance adjustments. With an active and supportive community, numerous resources, tutorials, and forums are readily available to assist users with questions and challenges.

Importantly, Visual Studio Code is both free and open-source, making it accessible to developers of all levels. Its lightweight design and powerful feature set have contributed to its widespread adoption, earning it a strong reputation in the coding community as a flexible and highly capable code editor.

5 System Requirement Study

A system requirement study, also known as a requirements analysis or requirements elicitation, is a crucial phase in the software development lifecycle where the needs and expectations of stakeholders are gathered, documented, and analyzed to define the scope and objectives of a project

5.1 User Characteristics

In our system, there will be two types of users

a. Admin: - In this system admin can only access the all-user's data who are previously used our system.
b. Normal User: - Normal user can visit our web application they can upload the resume, they will get the recommendations based on resume characteristics. They can delete the resume from the system.

5.2 System Main Module

It typically acts as the entry point for the application, coordinating the interaction between various modules, components, and layers within the system.

a. Resume Storing: User can upload the resume into system, we need to store that resume into our local system/server for future preprocessing.
b. PDF Extracting:User can only upload the PDF, now we need to processed PDF into number of Images, so this module will convert the PDF pages into Images
c. Text Extracting :After the converting PDF pages into Images, now our next step is to fetch all the text from the images, because for the NLP we

need to have the text, so this module will fetch the text from all the Images

d. Smart Recommendation: After the getting the text from the resume, we need to create recommendations for the particular data like giving data science skills and courses suggestions to person who have skills of data science. It will display the recommendations to the user.
e. Data Store: - After the giving recommendations to the user, our system will store the all user's data, it will be secure with our system. We are storing it for making our system better in future.

5.3 Functional Requirement

a. User need to visit the site.
b. User Upload Resume PDF into the System.
c. System will store the PDF.
d. PDF will be processed into Images.
e. Our System will fetch the Text from Images.
f. System will apply NLP on the text.
g. It will send the recommendation based on the resume to the system.
h. System will display the recommendations on the web applications.

5.4 Non-Functional Requirement

a. Accessibility: This can be used by any one because it is basic in the structure and usage.
b. Efficiency: This project efficiency as a first trial is not great by numbers but with more learning and hard work the app can be perfect and very efficient.
c. Scalability: This is now only on computers, so it requires a good internet

connection to work properly, still it will take a time to do pre-processing.

d. Security: We are not providing any Login/Registration for user, still user can upload the resume, resume will be deleted from the system, still user data will be secured with us for future training purpose.

5.5 Project Planning

We have built this project planning in Jira.

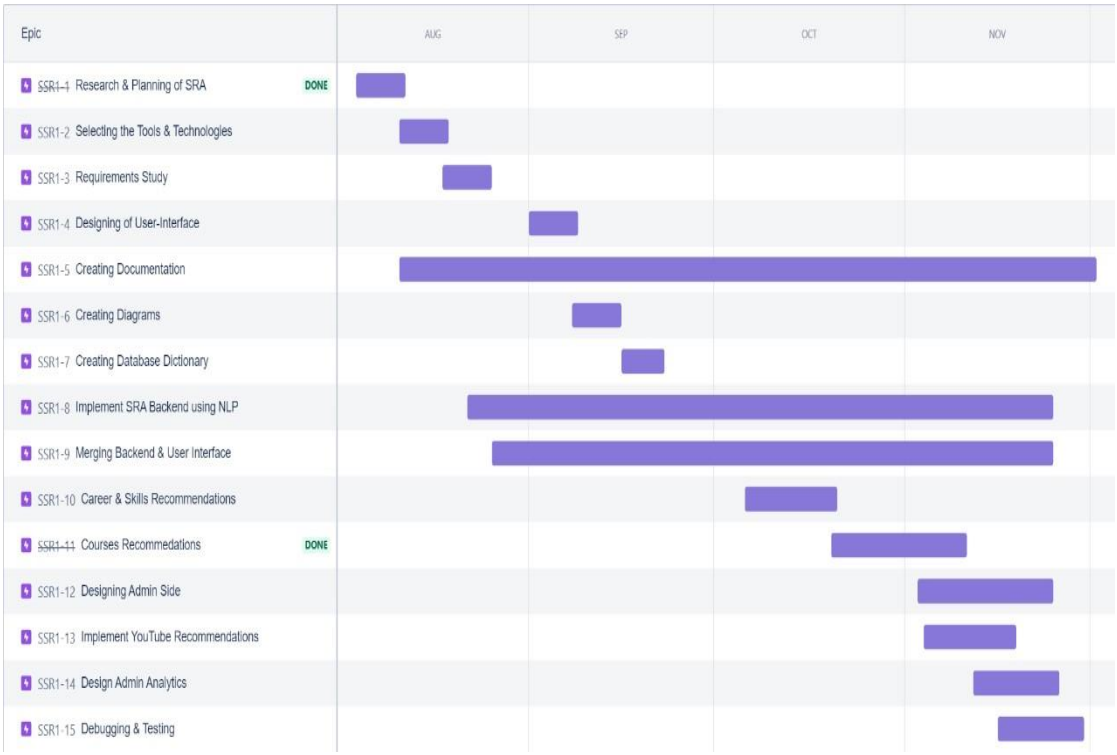


Figure 5.1 Project Planning

6 Methodology

In this chapter we are going to learn about working of Smart Resume Analyzer.

6.1 Working of Smart Resume Analyzer

1. We are going to see how actually our system is working behind, we have divided our work in separate tasks, let's understand each steps of it.

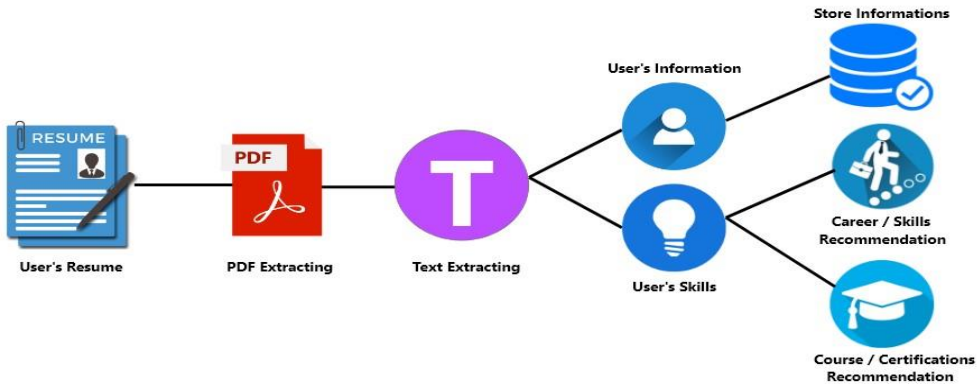


Figure 6.1 Working of Smart Resume Analyzer

PDF Extracting

PDF Extracting is a module that automatically retrieves the user's resume, provided that the resume is in PDF format. This module extracts the user's data from the resume.

Text Extracting

Text Extracting is a module that fetches the text information from the resume. This text data is used for language processing in further tasks, such as recommendations and fetching the user's personal information.

User's Data

After the text extraction, the next module involves fetching the user's information, such as their full name, contact details, email, mobile number, and skills, from the extracted text data.

Career/Skills Recommendations

Based on the user's current skills, this module provides career path and skills recommendations. For example, if the user has skills in Machine Learning, it will offer career, tools, and technology recommendations in that field.

Data Analytics

A new module called Data Analytics has been added. With a substantial amount of user data available, visualization becomes an effective technique for understanding patterns. We create visualizations, particularly pie charts, for the admin side to facilitate easy data comprehension.

YouTube Video Recommendation

Another new module called YouTube recommendations has been incorporated. The system recommends two types of videos to the user: one focusing on resume preparation and the other on interview preparation. These videos are directly scraped from YouTube without using any official API.

7 System Design

In this chapter, we will learn about the system designs, diagram and DFDs.

UML DIAGRAM

UML is an acronym that stands for Unified Modeling Language. Simply put, UML is a modern approach to modeling and documenting software. It is based on diagrammatic representations of software components. As the old proverb says: “a picture is worth a thousand words”. By using visual representations, we are able to better understand possible flaws or errors in software or business processes.

Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficacy.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. There are two broad categories of diagrams and they are again divided into subcategories

a. Structural Diagrams

b. Behavioral Diagrams

Structural Diagrams

The Structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and therefore stable. These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are

a. Class diagram
b. Object diagram
c. Component diagram
d. Deployment diagram

Behavioral Diagrams

Any System can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered. Behavioral diagrams basically capture the dynamic aspects of a system. UML has the following five types of behavioral diagrams.

a. Use case diagram
b. Sequence diagram
c. Collaboration diagram
d. Statechart diagram
e. Activity diagram

7.1 Use Case Diagram

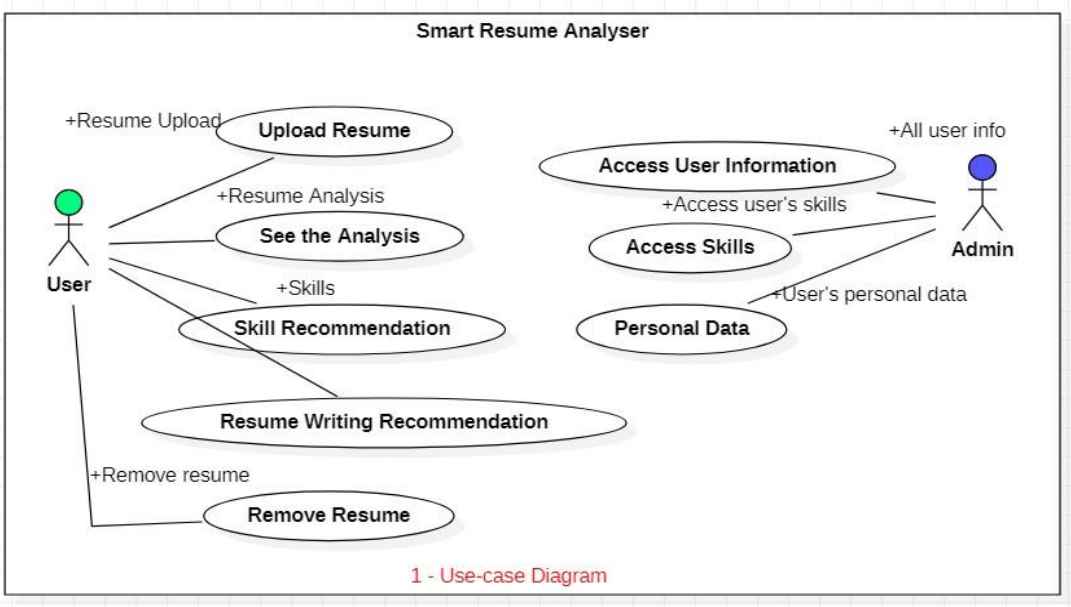


Figure 7.1 Use Case Diagram

a. In this diagram, we seen what the functionality can user and admin can use. We can see user and admin have the different access of the functionality.

b. A use case diagram at the simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

c. While a use case itself might drill into a lot of detail about every possibility, a use-case diagram can help provide a high-level view of the system. It has been said before that "Use case diagrams are the blueprints for your system". They provide a simplified and graphical representation of what the system must

actually do.

d. Due to their simplistic nature, use case diagrams can be a good communication tool for stakeholders. The drawings attempt to mimic the real world and provide a view for the stakeholder to understand how the system is going to be designed.

e. The purpose of the use case diagram is simply to provide the high-level view of the system and convey the requirements in laypeople's terms for the stakeholders.

f. Additional diagrams and documentation can be used to provide a complete functional and technical view of the system.

g. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

h. When the initial task is complete, use case diagrams are modelled to present the outside view. In brief, the purpose of use case diagrams can be said to be as follows –

    i. Used to gather the requirements of a system.
    ii. Used to get an outside view of a system.
    iii. Identify the external and internal factors influencing the system.
    iv. Shows the interactions among the requirements are actors.

7.2 Class Diagram

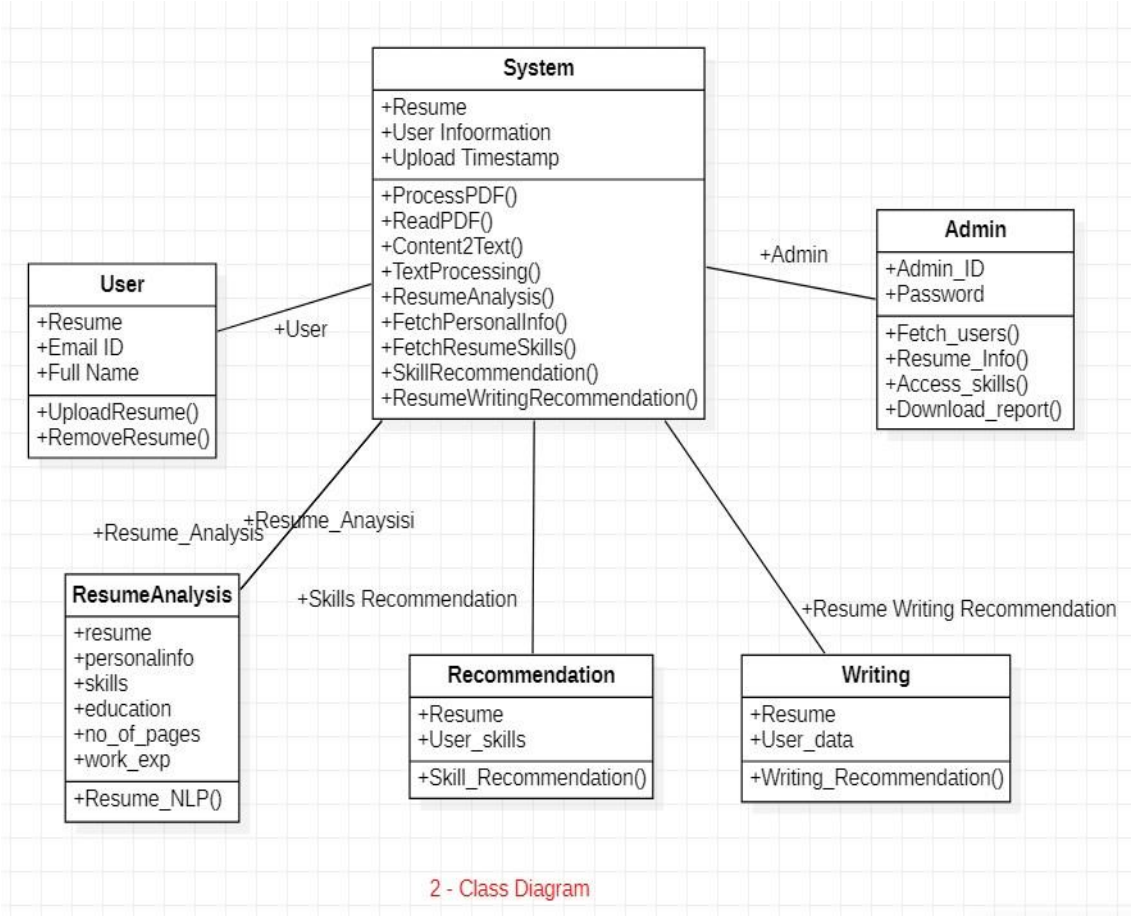


Figure 7.2 Class Diagram

a. In this diagram, we seen different classes and functions according to the functionality. We can see how system is associated with the user and admin functionality.

b. A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

c. The Class Diagram is the main building block of object- oriented modeling. It is used for general conceptual modeling of the structure of the application, and for detailed modeling translating the models into

programming code.

d. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

e. In the diagram, classes are represented with boxes that contain three compartments:

    i. The top compartment contains the name of the class. It is printed in bold and centered, and the first letter is capitalized.

    ii. The middle compartment contains the attributes of the class. They are left-aligned and the first letter is lowercase.

    iii. The bottom compartment contains the operations the class can execute. They are also left- aligned and the first letter is lowercase.

7.3 Sequence Diagram

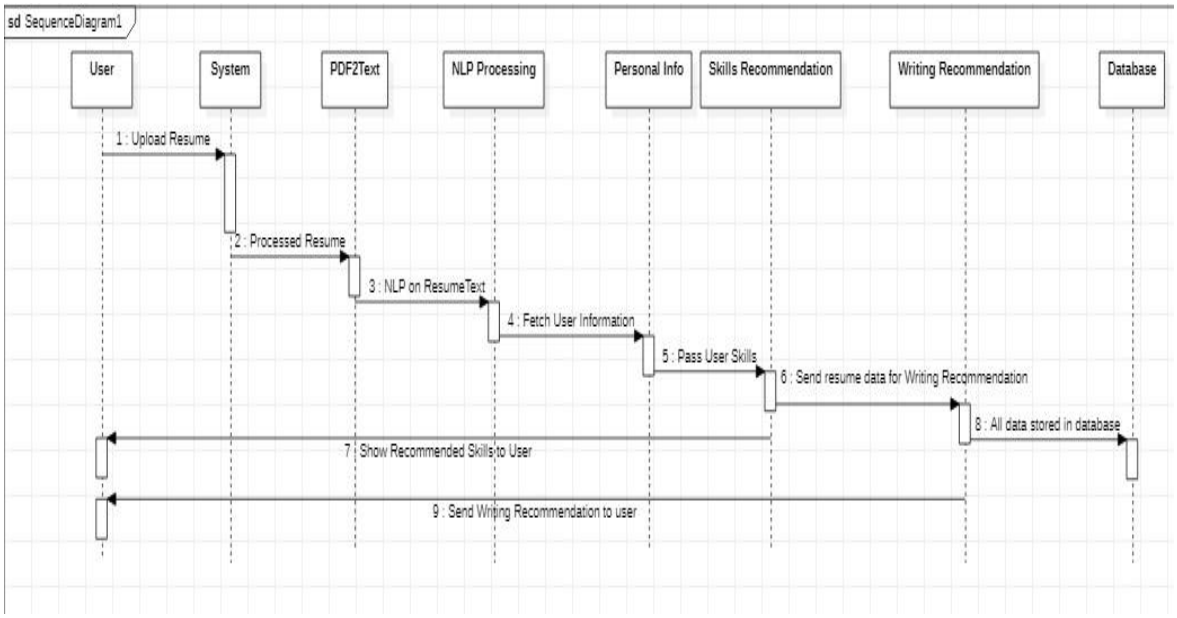


Figure 7.3 Sequence Diagram

a. In this diagram, we have seen the step-by-step procedure of Resume Processing. It is showing the sequence from Upload resume to get the Recommendations.

b. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

c. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development.

d. A sequence diagram shows, as parallel vertical lines (lifelines), different processes or objects that live simultaneously, and, as horizontal arrows,

the messages exchanged between them, in the order in which they occur.

e. This allows the specification of simple runtime scenarios in a graphical manner.

f. Messages, written with horizontal arrows with the message name written above them, display interaction. Solid arrowheads represent synchronous calls, open arrowheads represent asynchronous messages, and dashed lines represent reply messages.

g. Asynchronous calls are present in multithreaded applications, event-driven applications and in message-oriented middleware.

h. Activation boxes, or method-call boxes, are opaque rectangles drawn on top of lifelines to represent that processes are being performed in response to the message.

7.4 Activity Diagram

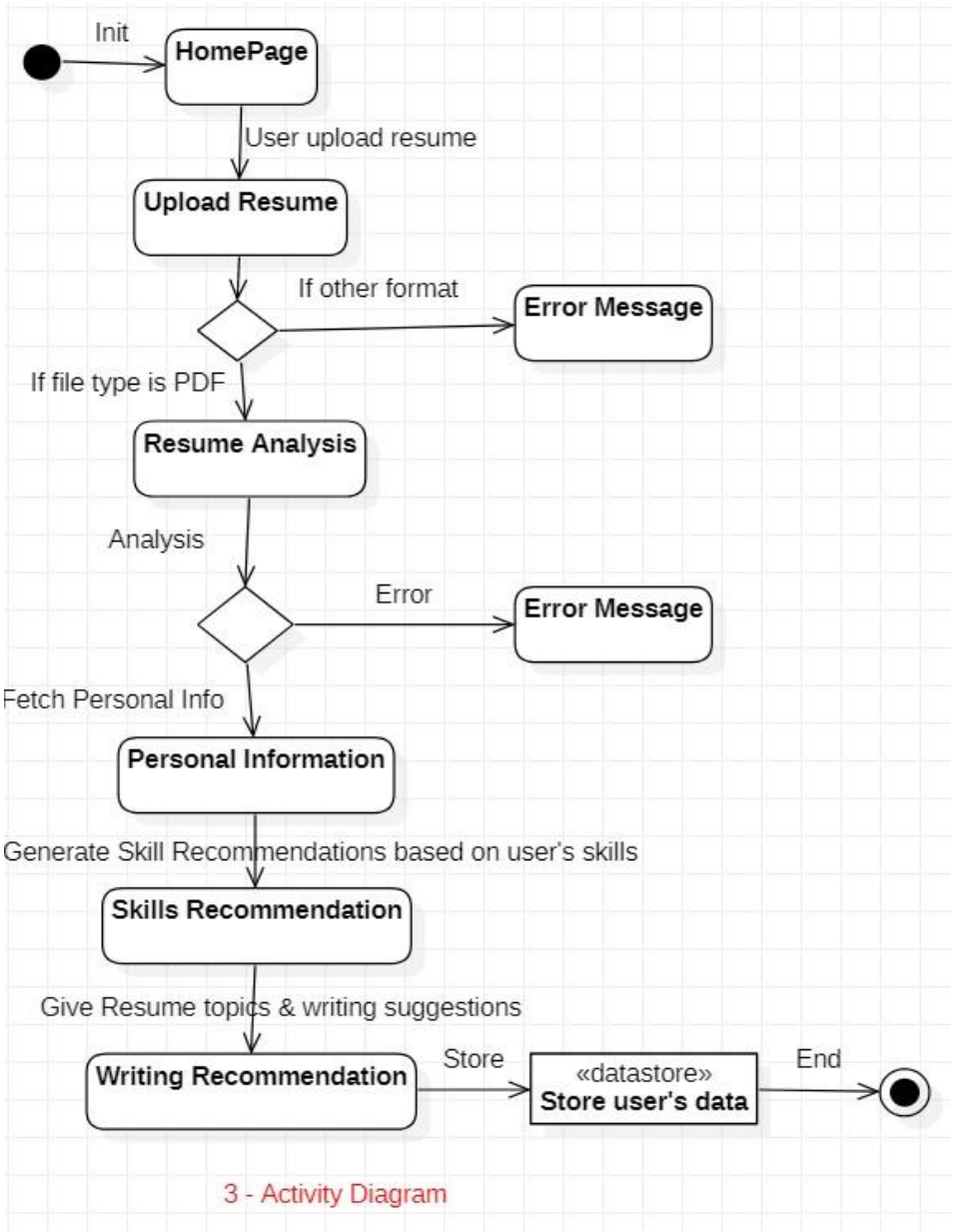


Figure 7.4 Activity Diagram

a. In this diagram, we can see that the activity of Smart Resume Analyzer, if everything works well, we will get the output, otherwise it will be error.

7.5 Data-Flow Diagram

A Data Flow Diagram (DFD) is a visual representation of the flow of data within a system, illustrating how data is input, processed, stored, and outputted. It's a powerful tool for analyzing and modeling the data flow and processes within a system.

7.5.1 Level-0 Diagram

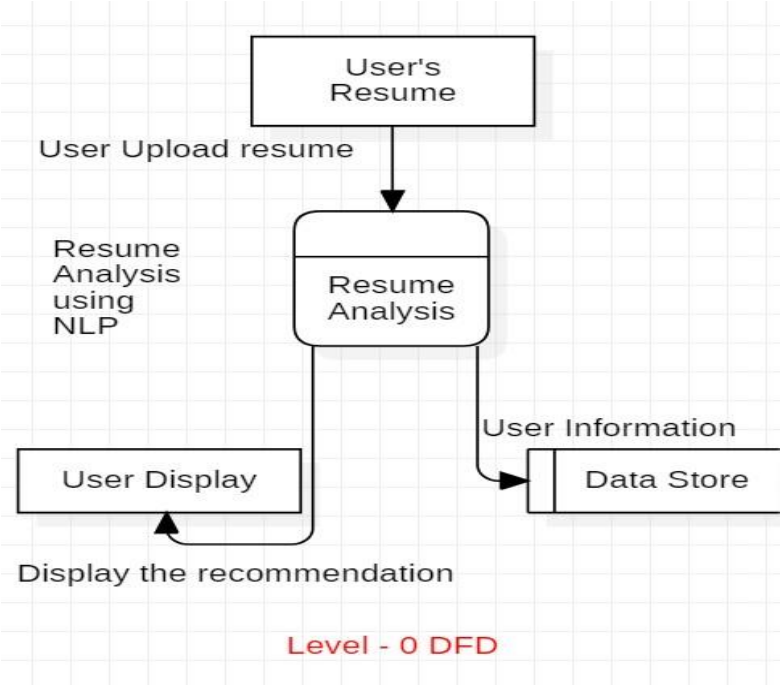


Figure 7.5 Level 0 Diagram

a. In this data-flow diagram, we can see that how system is associated with the Resume Analysis function and the database. We can see the data flow from resume to system, system to user and database.

7.5.2 Level-1 Diagram

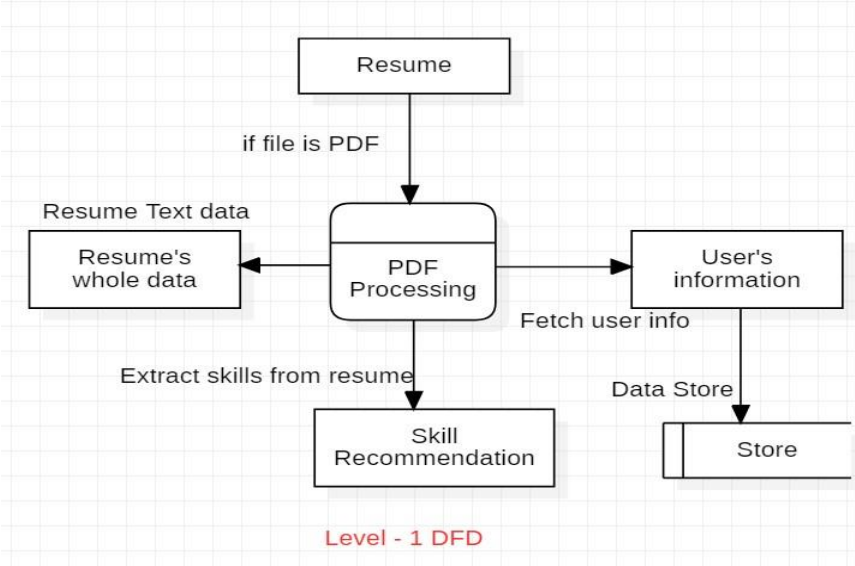


Figure 7.6 Level-1 Diagram

7.5.3 Level-2 Diagram

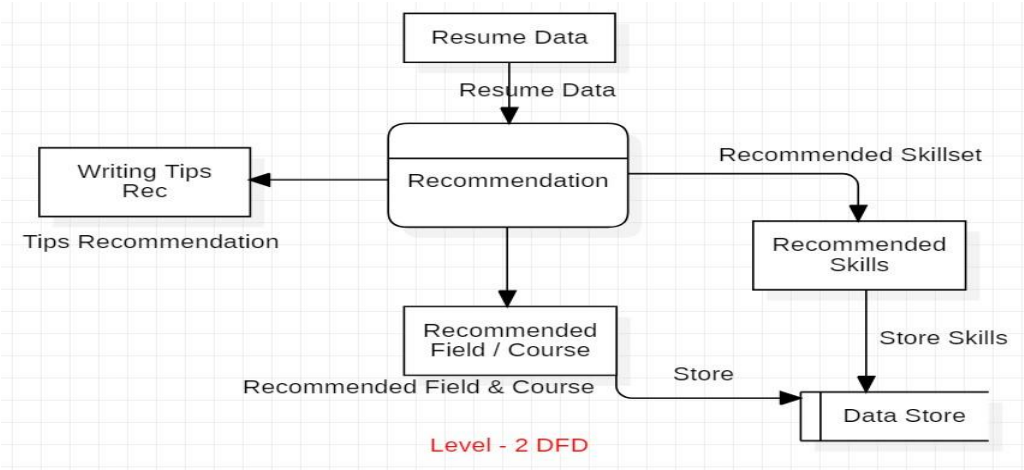


Figure 7.7 Level-2 Diagram

8 Database Dictionary

In this chapter we are going to understand the different tables of our database. A database dictionary, often referred to as a data dictionary or metadata repository, is a structured collection of descriptions about the data elements or entities within a database. It serves as a reference guide or documentation for understanding the structure, relationships, and attributes of the data stored in the database.

8.1 User Data Table

Table 8.1 User table

Column Name	Data Type	Description
ID	Auto Increment	Unique User ID
User Name	Varchar(50)	Full Name of user
Email ID	Varchar (50)	User Email ID
Resume score	Varchar (10)d	User’s auto generated score
Actual skills	Varchar (300)	User’s actual skills
Recommended skills	Varchar (300)	User’s Recommended skills
Recommended courses	Varchar (600)	Recommended courses

9 System Implementation

System implementation, in the context of a project, refers to the process of turning the design and specifications of the system into a working and operational reality. It involves translating the theoretical concepts and plans outlined in the project requirements and design phases into tangible software or hardware components.

9.1 Coding

The coding is the process of transforming the design of a system into a computer language format. This coding phase of software development is concerned with software translating design specification into the source code. It is necessary to write source code & internal documentation so that conformance of the code to its specification can be easily verified. Coding is done by the coder or programmers who are independent people than the designer. The goal is not to reduce the effort and cost of the coding phase, but to cut to the cost of a later stage.

9.2 Module Specification

In our system there will be two modules, admin & user. Admin will be only one. User can be multiple. Let see what user & admin can do.

Table 9.1 Module Specification

user	**admin**
Upload the Resume	Do Login
Check the own information	Check the all user's report
Check the skills/career recommendations.	Can export the report in CSV.
Check the certifications recommendations	Admin can see the Data An-alytics

9.3 Sample coding

The code which is mentioned below it is just main logic of giving the recommendations to the user. This code is only just functional part, not the full part with the backend.

PDF extracting code: This code is used for the PDF extraction; it will extract the Pdf of resume which is uploaded by the user.

```
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)
            print(page)
        text = fake_file_handle.getvalue()

    # close open handles
    converter.close()
    fake_file_handle.close()
    return text
```

Recommender Code: This code is used for the course recommender, it will fetch the user's skills, based on that skills it will give the recommendations.

```
def course_recommender(course_list):
    st.subheader("**Courses & Certificates🎓 Recommendations**")
    rec_course = []
    no_of_reco = st.slider('Choose Number of Course
Recommendations:', 1, 10, 4)
    random.shuffle(course_list)
    for c_name, c_link in course_list:
        c += 1
        st.markdown(f"({c}) [{c_name}]({c_link})")
        rec_course.append(c_name)
        if c == no_of_reco:
            break
    return rec_course
```

PDF Showing Code: This code is used to show the uploaded resume in the user interface, it simply allows to display the uploaded resume into the system.

```
def show_pdf(file_path):
    with open(file_path, "rb") as f:
        base64_pdf = base64.b64encode(f.read()).decode('utf-8')
    # pdf_display = f'<embed
src="data:application/pdf;base64,{base64_pdf}" width="700"
height="1000" type="application/pdf">'
    pdf_display = F'<iframe
src="data:application/pdf;base64,{base64_pdf}" width="700"
height="1000" type="application/pdf"></iframe>'
    st.markdown(pdf_display, unsafe_allow_html=True)
```

Insert user's Data Code: This code is used to insert the user data into our database

```
Definsert_data(name,email,mobile,timestamp,no_of_pages,reco_field,
cand_level,skills,recommended_skills,courses):
    DB_table_name = 'user_data'
    insert_sql = "insert into " + DB_table_name + """
    values (0,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s)"""
    rec_values = (name, email, str(mobile),
timestamp,str(no_of_pages), reco_field, cand_level,
skills,recommended_skills,courses)
    cursor.execute(insert_sql, rec_values)
    connection.commit()
```

Code URL

The GitHub repository hosts code and data for smart resume analyzer. This is about to analyze the resume of any person, based on resume our smart system will give the smart recommendation to the user. It is very easy to use and smart application, in this application user just need to upload the resume. It will automatically analyze the resume. After that it will give you the recommendations. Below is the provided URL for accessing the repository.

https://github.com/swathimakke/Smart-Resume-Analyzer.git

10 Testing

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Purpose Of Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests; each test type addresses a specific testing requirement. Testing and test design are parts of quality assurance that should also focus on bug prevention. A prevented bug is better than a detected and corrected bug. Testing consumes at least half of the time and work required to produce a functional program. History reveals that even well written programs still have 1-3 bugs per hundred statements. Testing is the process of executing a program with the aim of finding errors. To make our software perform well it should be error-free. If testing is done successfully, it will remove all the errors from the software.

Testing Strategies

In order to uncover the errors, present in different phases we have the concept of Level of testing. The Software testing has a prescribed order with the following list of software testing categories arranged in chronological order for our project testing and to generate test cases for output. These are the steps taken to fully test new software in preparation for marketing it:

Types of testing

a. **Unit testing** performed on each module or block of code during development. Unit Testing is normally done by the programmer who writes the code.

b. **Integration testing** done before, during and after integration of a new module into the main software package. This involves testing of each individual code module. One piece of software can contain several modules which are often created by several different programmers. It is crucial to test each module's effect on the entire program model.

c. **System testing** done by a professional testing agent on the completed software product before it is introduced to the market.

d. **Acceptance testing** - beta testing of the product done by the actual end users.

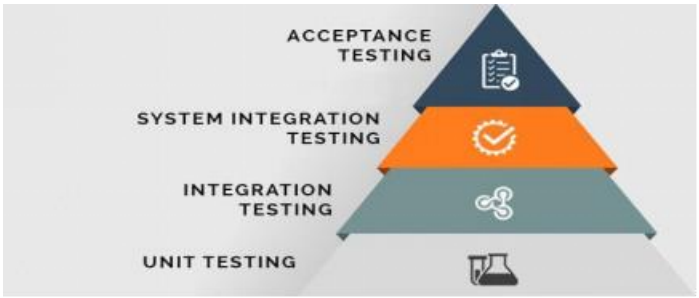


Figure 10.1 Types of Testing

Unit Testing

Unit testing focuses verification effort on the smallest unit of software i.e., the module. Using the detailed design and the process specifications testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test before the start of the integration testing begins. It has been seen that each activity class runs after its development using unit testing.

Unit testing is performed at the first stage of testing as it is performed first of all other testing processes. Unit testing is also known as white box testing. So it's generally performed by developers. In our Project Unit testing is commonly automated but may still be performed manually. Software Engineering does not favor one over the other but automation is preferred. A manual approach to unit testing may employ a step-by-step instructional document. The three Modules of Fake Face Detection GANs, Phase I & Phase II individually tested in terms of detection of defects. Used to test each one of those functions, behavior is tested.

Here the Unit Testing Techniques are mainly categorized into three parts which are Black box testing that involves testing of user interface along with input and output, White box testing that involves testing the functional behavior of the software application and Gray box testing that is used to execute test suites, test methods, test cases and performing risk analysis.

The Code coverage techniques used in our project under Unit Testing are listed below:

a. Statement Coverage
b. Decision Coverage
c. Branch Coverage

d. Condition Coverage

e. Finite State Machine Coverage

In this project, the unit testing was performed on each module separately to find any defects in the code. The testing performed was manual walkthrough in the code of each module whether the functional call was correct or not, whether the attributes of functions called were accurate or not. By unit testing we found that without proper execution of each individual block of code we cannot further execute the project. Each module should give an accurate result and then should be moved towards the next testing process. A piece of code cannot work if the file mentioned was not found in mentioned directory, by Unit Testing we can find such faults. Through unit testing of this project we found that each individual module or block of code was working without any faults.

Integration Testing

Integration testing is the testing where multiple modules are tested to verify if different pieces of the modules are working together as per expectation or not. In case of Integration testing multiple modules get integrated and are tested as a single module so testers focus more on integrated functionality rather on internal design of the application.

In our Project the Integration testing is performed after all modules get integrated are done with unit tested i.e. Integration testing is done after unit testing and before System testing.

Although each of our software module is unit tested, defects still exist for various reasons like below

1. A Module, understanding and programming logic may differ. To verify the software modules, work in unity.
2. At the time of module development, there are wide chances of change in requirements by the Users. These new requirements may not be unit tested.
3. Interfaces of the software modules with the dataset are sometimes erroneous.
4. External Hardware interfaces, if any, could be erroneous
5. Exception handling causes issues in execution sometimes.

Integration Test Case differs from other test cases in the sense it focuses mainly on the interfaces & flow of data/information between the modules. Here priority is to be given for the integrating links rather than the unit functions which are already tested.

Through integration testing we made sure each module was integrated properly and there were no errors in integrating the modules. The information was passed smoothly from one module to other modules without missing any data.

System Testing

System Testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications. Usually, the software is only one element of a larger computer-based system. Ultimately, the software is interfaced with other software/hardware systems. System Testing is actually a

series of different tests whose sole purpose is to exercise the full computer-based system.

In our project System Testing involves testing the software code for following

a. Testing the fully integrated modules including external peripherals in order to check how components interact with one another and with the system as a whole. This is to support an End-to-End testing scenario.
b. Verified with thorough testing of every input in individual modules to check for desired outputs.
c. Tested for user's experience and expectations with the integrated modules and System Application.

Here the entire software system is tested. The reference document for this process is a requirements document and the goal was to see if software meets its requirements.

Below we have listed types of system testing in our project typically used:

a. **Usability Testing-** it mainly focuses on the user's ease to use the application, flexibility in handling controls and ability of the system to meet its objectives of our project.
b. **Functional Testing –** to achieve functional completeness testing, Functional Testing involves trying to think of any possible missing functions. We might make a list of additional functionalities that a product could have to improve during functional testing.

10.1 Test Cases

Table 10.1 Test Case for SRA

Testcases	Input Data	Expected Output	Result
System able to fetch the user's information	User's Resume	User's Data	Pass
Page no of resume should generated.	User's Resume	Experience according to the resume pages	Pass
User skills should be generated	User's Resume	User's Skills	Pass
Recommended Career path	User's Skills	Recommend Career Path	Pass
Courses,Skills recommended.	User's Skill	Recommend skills & courses	Pass

Table 10.2 Test Case for Admin login

Testcases	Input Data	Expected Output	Result
Admin is able to login using Admin Credentials	Admin Id, Password	Admin redirects to admin panel.	Pass
Admin get the message when ID is correct but password is wrong	Admin Id, Password	Wrong password	Pass
Admin can see the all user's data into UI(Table)	User's Data	User's Data	Pass
Admin can See the Data Analytics	User's Data	Visualization	Pass

11 **Result Screenshots**

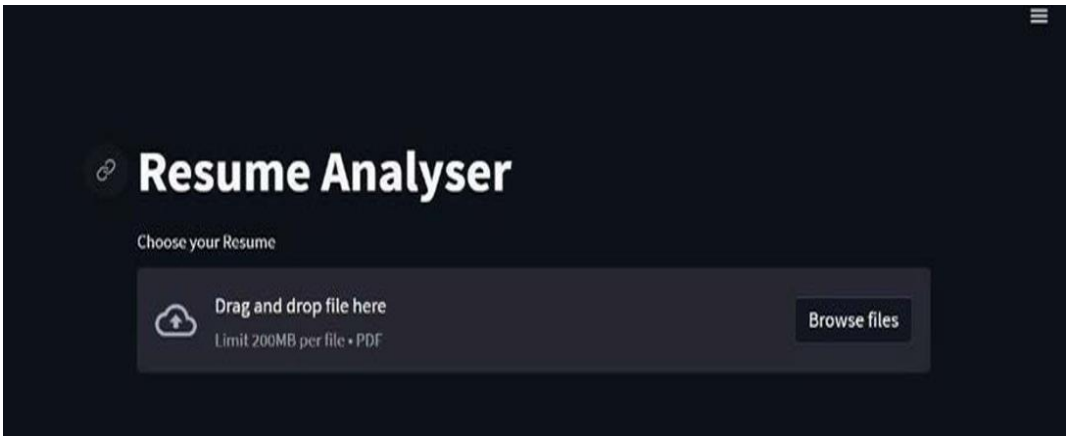
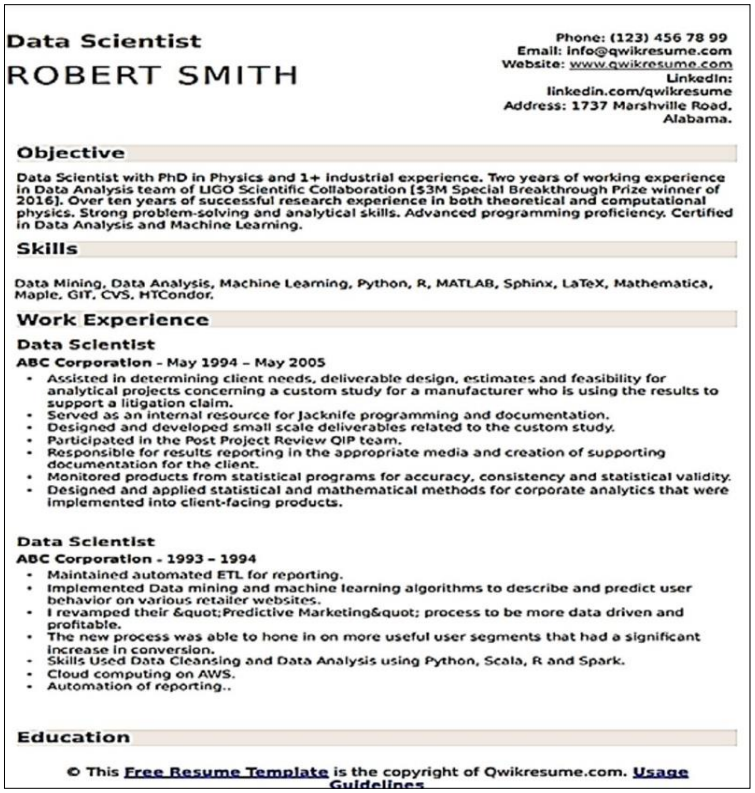


Figure 11.1 Landing Web Page of the Resume Screener



Data Scientist

ROBERT SMITH

Phone: (123) 456 78 99
Email: info@qwikresume.com
Website: www.qwikresume.com
LinkedIn:
linkedin.com/qwikresume
Address: 1737 Marshville Road, Alabama.

Objective

Data Scientist with PhD in Physics and 1+ industrial experience. Two years of working experience in Data Analysis team of LIGO Scientific Collaboration ($3M Special Breakthrough Prize winner of 2016). Over ten years of successful research experience in both theoretical and computational physics. Strong problem-solving and analytical skills. Advanced programming proficiency. Certified in Data Analysis and Machine Learning.

Skills

Data Mining, Data Analysis, Machine Learning, Python, R, MATLAB, Sphinx, LaTeX, Mathematica, Maple, GIT, CVS, HTCondor.

Work Experience

Data Scientist

ABC Corporation - May 1994 – May 2005

- Assisted in determining client needs, deliverable design, estimates and feasibility for analytical projects concerning a custom study for a manufacturer who is using the results to support a litigation claim.
- Served as an internal resource for Jacknife programming and documentation.
- Designed and developed small scale deliverables related to the custom study.
- Participated in the Post Project Review QIP team.
- Responsible for results reporting in the appropriate media and creation of supporting documentation for the client.
- Monitored products from statistical programs for accuracy, consistency and statistical validity.
- Designed and applied statistical and mathematical methods for corporate analytics that were implemented into client-facing products.

Data Scientist

ABC Corporation - 1993 – 1994

- Maintained automated ETL for reporting.
- Implemented Data mining and machine learning algorithms to describe and predict user behavior on various retailer websites.
- I revamped their "Predictive Marketing" process to be more data driven and profitable.
- The new process was able to hone in on more useful user segments that had a significant increase in conversion.
- Skills Used Data Cleansing and Data Analysis using Python, Scala, R and Spark.
- Cloud computing on AWS.
- Automation of reporting..

Education

Figure 11.2 Web page for displaying the resume after selecting for analysis

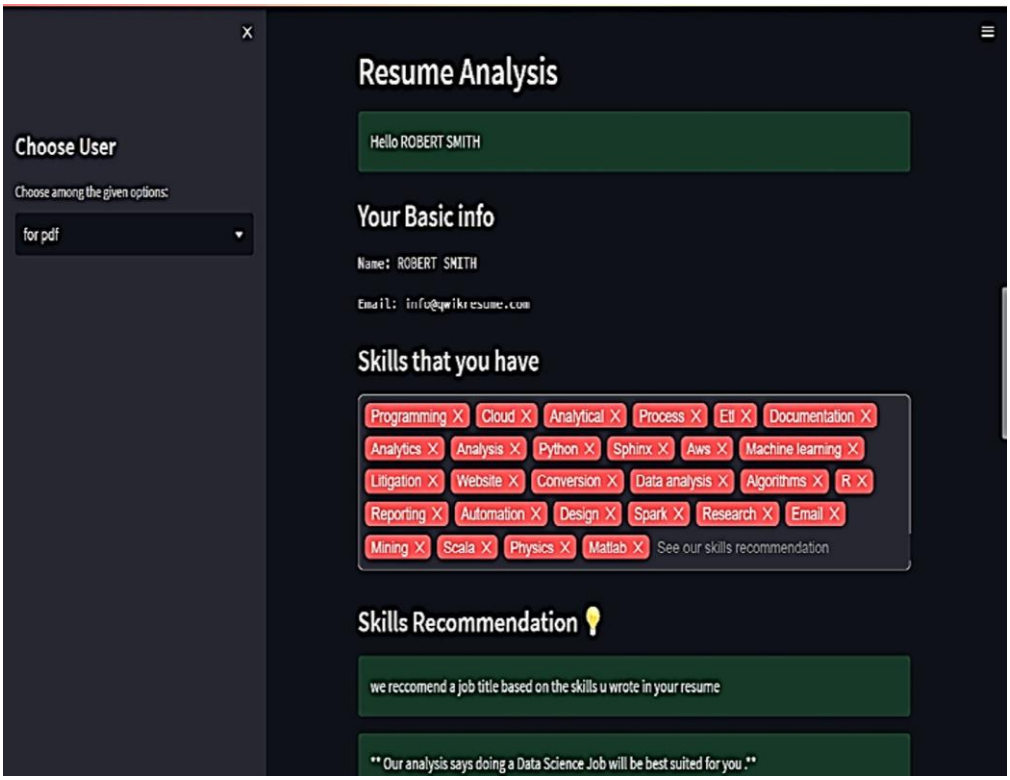


Figure 11.3 Skills will be Displayed based on your resume and additional skills will be recommended

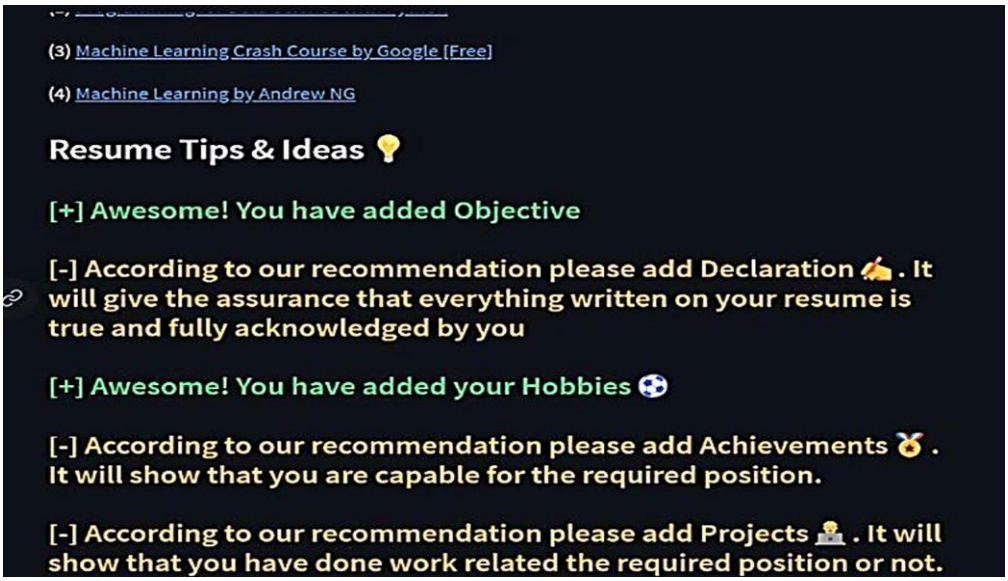


Figure 11.4 Tips

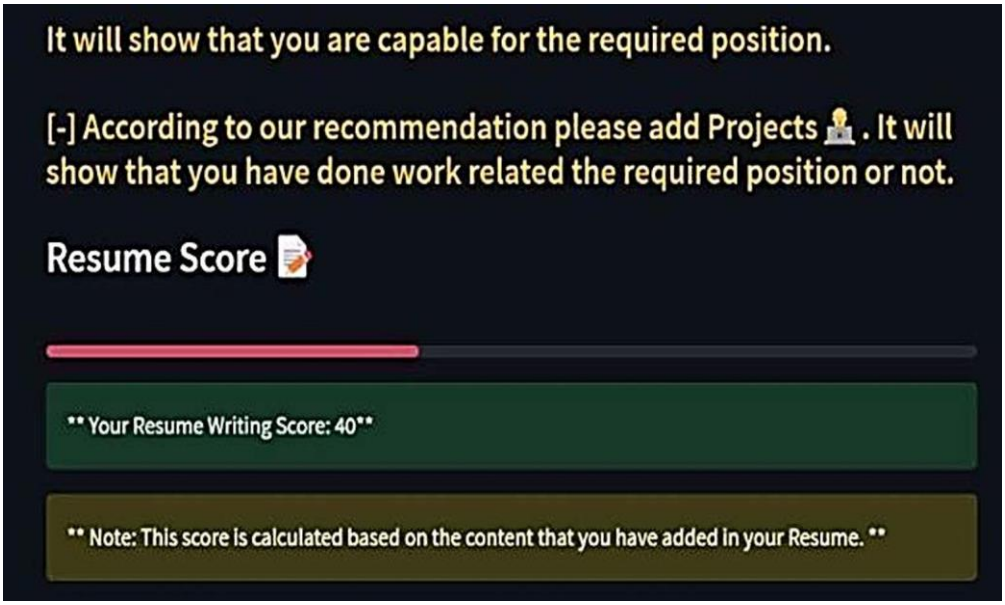


Figure 11.5 Score for the Resume

12 Risks and Challenges

a. Reliance on Keyword-based Analysis: One of the key risks associated with the smart resume analyzer application is its heavy reliance on keywords for assessing candidate skills. While keywords can provide a valuable indication of a candidate's expertise, there is a risk of oversimplification and misinterpretation. Since the application primarily focuses on keywords, there is a possibility that candidates who possess significant field knowledge but use fewer keywords to represent their skills might be inaccurately classified as beginners or intermediates. This limitation can lead to a lack of nuance in evaluating the true capabilities of candidates, potentially resulting in mismatches between job requirements and candidate qualifications.

b. Limited Applicability to Freshers: Another challenge associated with the smart resume analyzer is its restricted scope, as it is currently only designed to cater to freshers. This limitation can hinder its usefulness for organizations seeking candidates with experience beyond entry-level positions. By exclusively targeting freshers, the application may not adequately address the hiring needs of companies looking to fill mid-level or senior roles. This restriction can limit the application's market appeal and potential user base, posing challenges for its adoption and long-term viability.

c. Variability in Keywords and Industry-Specific Jargon: Different industries and job roles often have their own specific terminologies, acronyms, and jargon. This presents a challenge for the smart resume analyzer as it needs to accurately recognize and interpret industry-specific keywords in order to assess candidate skills effectively. Failure to account for these industry nuances can result in the

application misjudging a candidate's qualifications or misaligning them with job requirements. To mitigate this risk, continuous updates and maintenance of the application's keyword database would be necessary to keep up with evolving industry terminology.

d. Potential Bias and Inequality: Automated resume analysis systems, including the smart resume analyzer, can be susceptible to biases. The selection and weighting of keywords could inadvertently favor certain demographics or perpetuate existing inequalities in the job market. For example, if the system predominantly associates specific keywords with certain genders or ethnic backgrounds, it may unintentionally discriminate against qualified candidates from underrepresented groups. Ensuring fairness, diversity, and inclusivity in the application's algorithm and training data should be a priority to minimize such biases and promote equal opportunities.

e. User Adoption and Trust: A significant challenge for any new technology or application is gaining user adoption and establishing trust among potential users. Organizations may be hesitant to fully rely on an automated resume analyzer, especially if they have concerns about its accuracy and potential limitations. Building trust through rigorous testing, providing transparency about the system's limitations, and demonstrating the application's reliability can help address these concerns. Ongoing user feedback and continuous improvement of the system based on user input are also crucial for enhancing user adoption and satisfaction.

13 Conclusions

The purpose of this project is to learn the Natural Language Processing. We have research too many skills, tools and technologies for different types of IT jobs. Then we have done NLP processing that can recommend the Skills, Courses and career field. We learn to create one-page application development using Streamlit python. We have gathered the knowledge of Plotly for the visualization and Data analytics which is created in Admin Side. We have covered this project according to scheduled time. In this project we learn time management, Non-technical skills like documentations, diagrams drawing, Technical skills like NLP, Database handling, Visualization, Web scraping, Web development and many more things from this project.

14 Future Enhancements

a. Currently web applications are deployed locally, our future aim is to deploy them on the internet .

b. In the future, we will add more formats of resumes. Currently, the system only supports PDF format for uploading resumes.

c. This system currently works with a limited set of fields and recommendations, specifically designed for IT professionals. We plan to add more fields and data in the future to provide recommendations for all types of resumes.

d. We have achieved good accuracy in fetching user data, but sometimes the displayed data fetched from the PDF is incorrect. We will improve this in the future.

e. In the mobile view of the application, there are occasional UI lags. We will address this issue soon.

15 References

[1] Mr. Chirag Dariyani, Gurmeet Singh Chhabra, Harsh Patel, Indrajit Chhabra. “An Automated Resume Screening Using Natural Language Processing And Similarity.” *Ethics and Information Technology*.

[2] Shubham Bhor, Vivek Gupta, Vishak Nair, Harish Shinde, Mansi Kulkarni. “A Resume Parser Using Natural Language Processing Technique.” *International Journal of Research in Engineering and Science (IJRES)*.

[3] Satyak Sanyal, Souvik Hazra, Neelanjan Ghosh, Soumyashree Adhikary. “Resume Parser with Natural Language Processing.” *2017 IJESC*.

[4] Pradeep Roy, Sarabjeet Chaudhary, Rocky Bhatia. “A Machine Learning Approach for Automation of Resume Recommendation System.” *ICCIDS 2019*.

[5] Streamlit Documentation. *Documentation from Streamlit Developer Community*.

[6] OpenCV Documentation. *Documentation from OpenCV Developer Community*.

[7] Pillow Documentation. *Official documentation from PIL Developer Community*.

[8] NLTK Documentation. *Official documentation from NLTK Developer Community*.

[9] PyResParser Documentation. *Official documentation from PyResParser Developer Community*.