

CURRICULUM VITAE ANALYSIS USING MACHINE LEARNING ALGORITHM

A Project Report submitted in the partial fulfillment of the requirements for the
award of the Degree of

BACHELOR OF TECHNOLOGY
In Computer Science and Engineering

Submitted By
K.GAYATHRI VENKATA AASRITHA
(Regd. No:20NE1A0571)

Under The Esteemed Guidance Of
Mrs.J.LAKSHMI M.Tech,(Ph.D)
Assoc. Professor



Department of Computer Science & Engineering
TIRUMALA ENGINEERING COLLEGE

(Approved by AICTE & Affiliated to JNTU, KAKINADA, Accredited by NAAC A+ & NBA)

Jonnalagadda, Narasaraopet, PALNADU (Dt.), A.P.

(2020-2024).

TIRUMALA ENGINEERING COLLEGE

(Approved by AICTE & Affiliated to JNTU KAKINADA, Accredited by NAAC A+ & NBA)

Jonnalagadda, Narasaraopet-522601, Guntur (Dist) A.P

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that the project report entitled “**CURRICULUM VITAE ANALYSIS USING MACHINE LEARNING ALGORITHM**” is the bonafied work carried out by **K.Gayathri Venkata Aasritha.(20NE1A0571)** in partial fulfillment of the requirements for the award of “Bachelor of Technology” degree in the Department of CSE from J.N.T.U. KAKINADA during the year 2020- 2024 under our guidance and supervision and worth of acceptance of requirements of the university.

PROJECT GUIDE

Mrs.J.Lakshmi M.Tech,(Ph.D)

HEAD OF DEPARTMENT

Dr. N. GOPALA KRISHNA M.Tech,Ph.D,MISTE

PROJECT COORDINATOR

Mr. S. Anil Kumar M.Tech.(Ph.D)

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We wish to express our thanks to various personalities who are responsible for the completion of the project. We are extremely thankful to our beloved chairman **Sri. Bolla Brahma Naidu**, our secretary **Sri. R. Satyanarayana**, Who took keen interest in our every effort throughout this course. We owe our gratitude to our principal sir **Dr.Y.V.Narayana**^{M.E.,Ph.D}, **FIETE** for his kind attention and valuable guidance throughout the course.

We express our deep felt gratitude to our **H.O.D Dr. N. Gopala Krishna** and **Mr.S.Anil Kumar**, coordinator of the project for extending their encouragement. Their profound knowledge and willingness have been a constant source of inspiration for us throughout the project work.

We wish to express our sincere deep sense of gratitude to our guide **Mrs.J. Lakshmi**, for significant suggestions and help in every respect to accomplish the project work. His persisting encouragement, everlasting patience and keen interest in discussions have benefited us to the extent that cannot be spanned by words to our college management for providing excellent lab facilities for completion of project within our campus.

We extend our sincere thanks to all other teaching and non-teaching staff of department of CSE for their cooperation and encouragement during our B. Tech course.

We have no words to acknowledge the warm affection, constant inspiration and encouragement that we received from my parents. We affectionately acknowledge the encouragement received from my friends and those who involved in giving valuable suggestions and clarifying our doubts which had really helped us in successfully completing my project.

BY:

K.GAYATHRI VENKATA AASRITHA
20NE1A0571

INDEX

	<u>PAGE NO</u>
1. INTRODUCTION	1-3
2. LITERATURE SURVEY	4-9
3. SYSTEM ANALYSIS	10-21
3.1 Existing System	11-12
3.2 Proposed System	13-15
3.3 System Architecture	16-17
3.4 Modules and Description	18
3.5 Feasibility Study	19
3.5.1 Economical Feasibility	19
3.5.2 Technical Feasibility	19
3.5.3 Social Feasibility	20
3.6 Requirement Specification	21
3.6.1 Hardware Requirements	21
3.6.2 Software Requirements	21
4. SYSTEM DESIGN	22-29
4.1 Input and Output design	23-24
4.2 UML diagram	25
4.2.1 Use case Diagram	26-27
4.2.2 Sequence Diagram	28
4.2.3 Activity Diagram	29

5. IMPLEMENTATION	30-55
5.1 Software Environment	31-36
5.1.1 Java technology	31-33
5.1.2 Developing J2ME Applications	33
5.1.3 Design Considerations	34-36
5.2 Source Code	37-52
5.3 Output Screens	53-55
6. TYPES OF TESTING	57-63
6.1 Test Strategies	56-59
6.2 Test Objectives	60-61
6.3 Test Cases	62-63
7. CONCLUSION	64-65
8. FUTURE ENHANCEMENTS	66-67
9. BIBLIOGRAPHY	68-70

ABSTRACT

ABSTRACT

We describe a system called CV analysis machine, that processes a set of given free-form textual resumes (in English for IT domain), creates a standardized profile for each candidate, and for a given job description, identifies a ranked shortlist of k candidates, along with a matching score for each. The resume scoring function is hand-crafted, hierarchical, and uses domain-knowledge from recruitment experts. We describe a simple neural-network system that automatically learns some weights used in the scoring function, based on feedback about whether the candidate with their grades on each portion and also calculate overall grade of each candidate and also find all grades at a time. Companies often receive thousands of resumes for each job posting and employ dedicated screeners to short list qualified applicants. In this paper, we present PROSPECT, a decision support tool to help these screeners shortlist resumes efficiently. Prospect mines resumes to extract salient aspects of candidate profiles like skills, experience in each skill, education details and past experience. Extracted information is presented in the form of facets to aid recruiters in the task of screening. We also employ Information Retrieval techniques to rank all applicants for a given job opening. In our experiments we show that extracted information improves our ranking by 30% there by making screening task simpler and more efficient.

INTRODUCTION

1. INTRODUCTION

As far as employment is considered, selecting the right candidate for the recruitment process from a vast pool of candidates has been a fundamental issue. Conducting personality and various technical eligibility evaluation tests, interviews, and group discussions have been traditional techniques. Due to inception of social media, much more important information about employees is exposed to their online handles. Generally, such information is unnoticed by the recruiters. Aptitude test followed by the interview is traditional practices for the recruitment process. These traditional practices are very much time-consuming, and may result in unfair choices of candidate. As compared to traditional recruitment process, if an online selection process is conducted, then a fair selection of the candidate is possible.

Personality is the most important factor which reflects an individual, which keeps on varying. Tackling them is a tedious task for which we have implemented an approach to identify the personality and also provide with the recommendation. This will enable a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy, which can be legally justified. System will rank the experience and key skills required for particular job position. Than system will rank the CV's based on the experience and other key skills which are required for particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy. This system will focus not only in qualification and in experience but also focuses on other important aspects, which are required for particular job position. This system will help the human resource department to select right candidate for particular job profile, which in turn provide expert workforce for the organization.

In this paper, we propose a machine learning based method to check a candidate's aptitude and personality score. The personality of the candidate would be identified by using two metrics, first is aptitude /personality test and second CV analysis. The administrator is responsible to design, update or drop the questions and has the complete control to customize. aptitude/personality questions as per organization requirements. Further, three categories of questions are added in the aptitude test which includes quantitative, verbal and logical type questions. After the aptitude test, the personality test is carried out so that candidate's personality would be tested [6]. The decision can be made on the basis of the test outcome.

Finally, the score of the test is displayed and the decision of the candidate

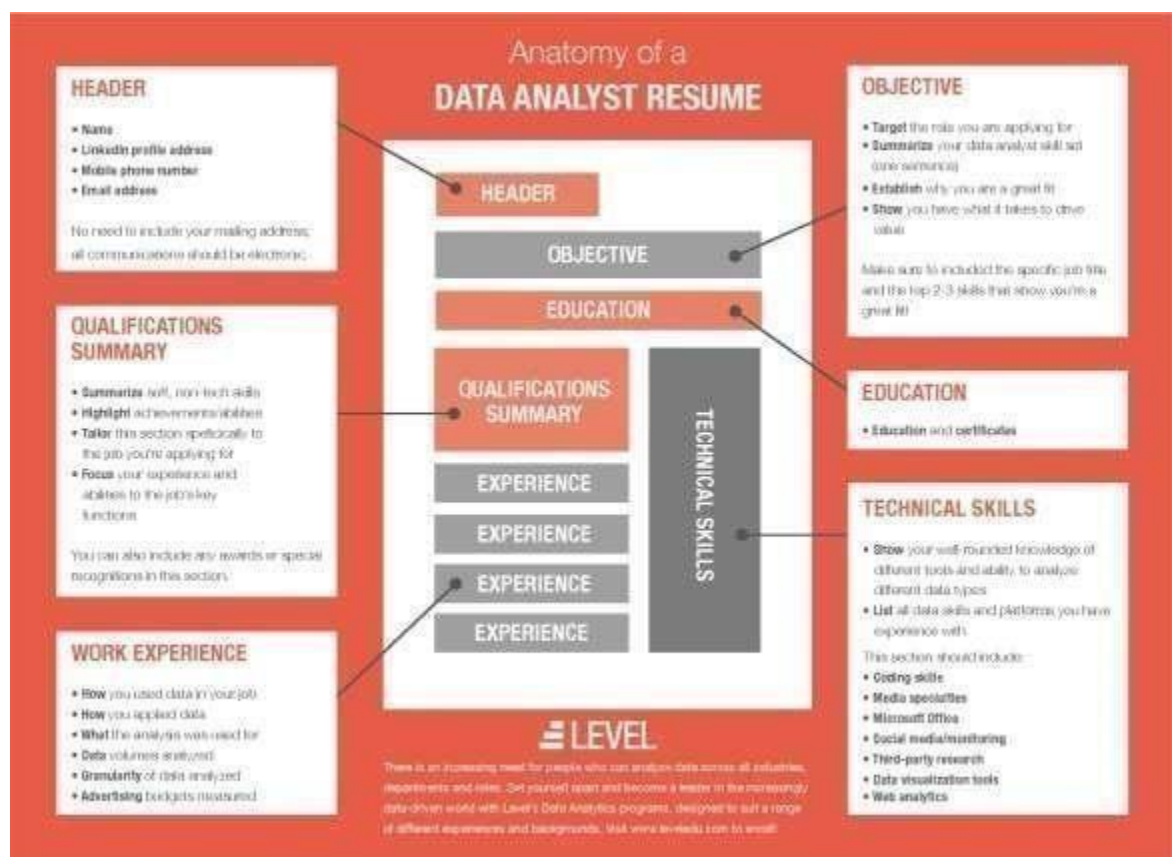


Fig 1.1 : Resume modules

LITERATURE SURVEY

2. LITERATURE SURVEY

However, much of the resume research literature is dated and there is a lack of an organizing framework regarding future resume-related research. Thus, the purpose of the current paper is to provide: (1) a synthesis of the historical empirical research literature through the lens of the advice that has accumulated to date; and (2) an organizing framework containing future research questions that need to be investigated in order to continue moving the literature forward. The current paper will be of use to job applicants, business communication instructors, and researchers. Keywords: resume, cover letter, job applicant, job search, business communication, personnel selection

1. Introduction

It has long been established that first impressions have a substantial impact on the interpretation of subsequent information (e.g., **Anderson & Barrios**, 1961). In a business context, the first impression of a job applicant has traditionally been made through the resume (e.g., **Knouse**, 1989; **Roderick & Jelley**, 1992.

Stanley-Weigand, 1991) and, despite many technological advancements since the initial resume-related publications began to appear in the research literature, the resume remains a common application component (e.g., **Catano, Wiesner, & Hackett**, 2016). Thus, the importance of job applicant resumes in terms of business communication cannot be understated; as one earlier resume communication scholar professed: “why teach anything about communicating within the organization unless we teach about how to get into the organization?” (**Stull**, 1978, p. 8). Business communication instructors often teach a module on resume and cover letter writing.

During this module, it is not uncommon for undergraduate and graduate students to mention the conflicting advice that they have received from the institution’s career center, their other instructors, their friends, their family members, the popular press, and so on. Thus, it would be beneficial to have a current resource that synthesizes the extant empirical research literature in order

Mullainathan, 2004; Dipboye, Arvey, & Terpstra, 1977; Dipboye, Fromkin, & Wiback, 1975; Marlowe, Schneider, & Nelson, 1996; McIntyre, Moberg, & Posner, 1980; Muchinsky & Harris, 1977; Oliphant & Alexander, 1982; Renwick & Tosi, 1978) is not incorporated into the below discussion. Clearly, it has been established that the following personal information components should not be included in resumes: height, weight, race, religion, birth date, marital status, number of dependents, physical/health status, photographs, and social security number (**Hutchinson & Brefka, 1997**). Moreover, consistent with the empirical research-based mandate of the current paper, publications that involve merely opinions without supporting research evidence (e.g., **Dittman, 1983; Stanley-Weigand, 1991**) are also not incorporated into the below discussion. The below discussion will involve a comprehensive integration of the most impactful, systematic, and empirical research available in order to see how the following questions have been historically answered in the literature: What information should be included in the resume? How long should the resume be? In what order and style should the resume content be arranged? What is the best available advice for cover letter writing? What other advice is there for resume and cover letter writing? Based on the information and advice that has accumulated to date, a framework for future research needs will be offered. Overall, the work presented in the current paper will be of use to job applicants, business communication instructors.

Researchers interested in updating and advancing the resume research literature. 2. What Information Should Be Included in the Resume? Compared with an application form, job applicants can choose what information to include in their resumes. Some of the options for categories to include are: (a) personal information; (b) personal opening, job objective, career objective, and summary of qualifications; (c) education; (d) work experience; (e) references; (f) scholarships, awards, and honors; (g) hobbies, interests, and extracurricular activities; and (h) willingness to relocate and travel. Next, the empirical research

literature regarding the information that should be included in the resume for each of the aforementioned categories is reviewed. 2.1 Personal Information A large and well-established body of research has suggested that the applicant's name, address, and phone number should definitely be included in the resume (**Hornsby & Smith, 1995; Hutchinson, 1984; Hutchinson & Brefka, 1997; Mansfield, 1976;**

Wells, Spinks, & Hargrave, 1981). Although not focused on stylistic resume issues, a recent study by **Burns, Christiansen, Morris, Periard, and Coaster (2014)** has provided support for the inclusion of a school email address over a personal email address as their sample of human resource professionals provided more favorable judgments for resumes that included email addresses containing ".edu". The advice to include a school email address over a personal email address is especially applicable for student applicants and recent graduates but may also have implications for additional applicants. Specifically, future research should address whether more professional email addresses (e.g., givenname.surname@emailprovider.com) are rated more favorably than less professional email addresses. 2.2 Personal Opening, Job Objective, Career Objective, and Summary of Qualifications Although recent research has provided support for the notion that there is no need for a personal opening in a resume (**Burns et al., 2014**), including a job objective and/or a career objective has traditionally been found to be important information to include in a resume (**Harcourt & Krizan, 1989; Harcourt, Krizan, & Merrier, 1991; Hornsby & Smith, 1995; Hutchinson, 1984; Hutchinson & Brefka, 1997; Schramm & Dortch, 1991**). **Harcourt and colleagues'** (1991) sample of 212 campus recruiters demonstrated a preference for a career objective over a job objective or a combined career and job objective. However, future research is needed to more conclusively demonstrate the impact of including a career objective, a job objective, or a combined career and job objective especially considering the technological advancements since some of this research was conducted.

Lastly, a summary of qualifications may also be important to include in a resume (**Harcourt & Krizan**, 1989; **Harcourt et al.**, 1991); however, future research is also needed to determine whether including a summary of qualifications is effective if a career objective, a job objective, or a combined career and job objective has already been included. jedp.ccsenet.org Journal of Educational and Developmental Psychology Vol. 7, No. 1; 2017 171 Overall, the best available advice for resume writers is to avoid a personal opening and to instead include an objective or a summary of qualifications.

2.3 Education

Both early and more recent resume research agree that resumes should include formal educational qualifications including information regarding the degree or designation as well as the major, minor, and, if applicable, the expected date of graduation (**Burns et al.**, 2014; **Feild & Holley**, 1976; **Harcourt et al.**, 1991; **Holley, Higgins, & Speights**, 1988; **Horn**, 1988; **Hutchinson**, 1984; **Hutchinson & Brefka**, 1997; **Mansfield**, 1976; **McNeilly & Barr**, 1997; **Schramm & Dortch**, 1991; **Spinks & Wells**, 1987, 1993; **Su, Lorgnier, Yang, & Oh**, 2015; **Wellset al.**, 1981).

A less intuitive piece of advice for resume writers, based on a national study of practicing Society for Human Resource Management professionals, is to include Grade Point Average (GPA) data including GPA within major and possibly also overall GPA (**Ray, Stallard, & Hunt**, 1994). In fact, a convincing body of early and more recent research has supported the inclusion of GPA data in the resume (e.g., **Brown & Campion**, 1994; **Burns et al.**, 2014; **Campion**, 1978; **Chen, Huang, & Lee**, 2011; **Feild & Holley**, 1976; **Harcourt & Krizan**, 1989; **McNeilly & Barr**, 1997; **Schramm & Dortch**, 1991) and the preference for this type of information increased since the **Hutchinson** (1984) study (**Hutchinson & Brefka**, 1997). **Hutchinson** and **Brefka's** (1997) sample of personnel administrators in fortune 500 organizations demonstrated a preference.

SYSTEM ANALYSIS

3.SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

The candidates will register themselves with the required details in the CV form and upload the documents for verifying the authenticity of the information provided in the CV registration form. The candidates can then view the requirements and details of the job as specified by the admin or the recruiter. In order to apply for a job the candidates needs to be eligible for that job by satisfying all the requirements as stated by the recruiter. If a candidate applies for a specific job, the system checks whether the candidate meets all the requirements/parameters as specified by the recruiter. If the candidate satisfies the requirements/parameters, then the candidates request for that job would be accepted else the system denies the candidate's request for that job.

Candidate can also give an online test, which will be conducted on personality questions as well as aptitude questions. After completing the online test, candidate can view their own test results in graphical representation with marks. The system would then derive and rank the candidates who were eligible for the job. The rank of each candidate acts as a score of how well the candidate's profile meets the specifications of the recruiters as well as cumulative score of the aptitude test. The recruiter could also analyse the personality of the candidate based on the result of the personality test. So, based on CV, aptitude test and the personality test the candidate would be selected.

DIS ADVANTAGES

- This system requires large memory space as it stores data related to CV's.
- Requires an active internet connection.
- May provide inaccurate results if data not entered properly.
- To verify all the candidates forms it takes more time.
- There is a huge workload on the human resource department to select the right candidate for a particular job profile which in turn would provide experts workforce for the organization from a large pool of candidates.

3.2. PROPOSED SYSTEM:

Problem Statement:

There is a huge workload on the human resource department to select the right candidate for a particular job profile which in turn would provide experts workforce for the organization from a large pool of candidates.

- Gather a dataset of CV's along with their associated profile.
- Extract relevant data features from the CV data, such as education, work experience, skills etc.
- Choose a suitable Deep Learning algorithm for the task.
- Split the dataset into training, validation, and testing sets. Train the deep learning model on the training data using techniques like mini-batch gradient descent and back propagation.
- Evaluate the trained model on the testing set using evaluation metrics such as accuracy, F1 score, precision.
- Continuously monitor the deployed model's performance and update it as needed with new data.

DESIGN ALGORITHM

In this section we state the machine learning algorithm (TF-IDF) for CV analysis. The TF-IDF [11] Algorithm is used to find out the important keywords in a document/CV. Below, we give the working of TF-IDF in detail.

Step 1: Calculate TF (Term Frequency)

Term Frequency (TF) - Number of times a keyword appeared in a document is calculated by Term Frequency.

$TF(\text{'keyword'}) = \text{number of times 'keyword' appears in document} / \text{Total number of keywords in the document}$.

Here, the term 'keyword' signifies any job specific skill which the algorithm is searching for.

Step 2: Calculate IDF (Inverse Document Frequency) value.

The problem of rare and frequent words is resolved.

This helps our system to give more priority to the required word or skills.

IDF sets the log value=1 for the required CV as per skill sets and log value=0 for the unwanted CV.

$IDF(\text{'keyword'}) = \log(\text{total number of CV} / \text{Number of document with term 'keyword'})$

Step 3: Calculate TF-IDF weight

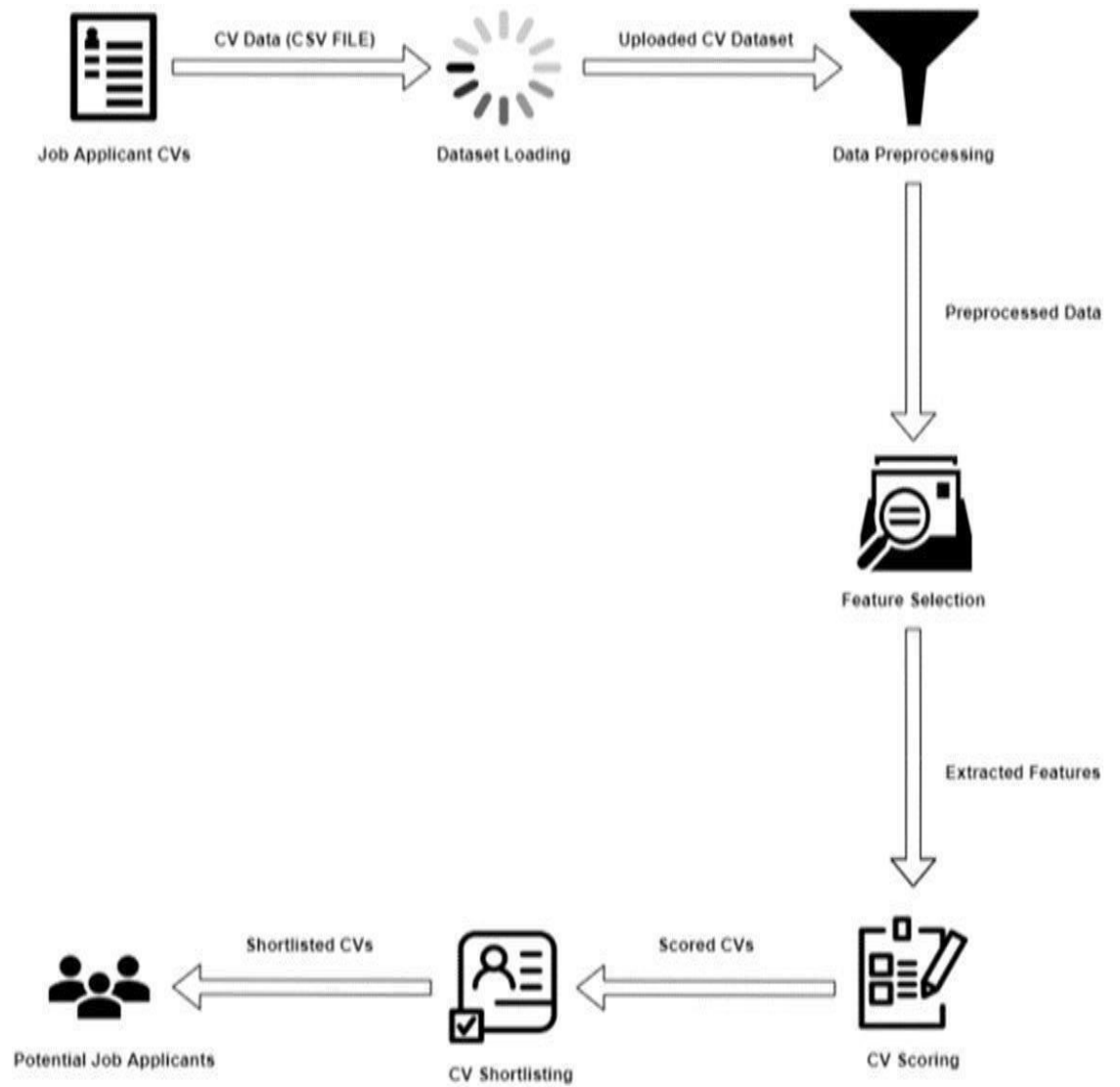
Weight = $TF(\text{'keyword'}) * IDF(\text{'keyword'})$

Higher the weight, more relevant is the CV and lower the weight, less or no relevance of the CV for the selection process. This step returns the CV with highest and lowest weight values which is further useful for classification.

ADVANTAGES:

- The proposed system will enable a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy. This can be legally justified.
- System will rank the experience and key skills required for a particular job position than system will rank the CV's based on the experience and other key skills which are required for particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy.
- This system employs a machine learning algorithm namely "Logistic Regression" which helps to choose fair decisions to recruit a suitable candidate.
- It has the potential to transform research and assessment in personality psychology. Algorithms can handle vast datasets, including thousands of attributes, without succumbing to collinearity issues. Moreover, ML algorithms are highly efficient in recognizing patterns in datasets that humans cannot even perceive.
- It does not consume more time to identify the personalities.
- This predictive personality makes the recruiters understand if a candidate will be a top performer and fit to the culture of the company easily.

3.3 SYSTEM ARCHITECTURE :



The architectural workflow in a sequential process is depicted in the above figure. Workflow shows the interaction among the entities like Job Applicant CV's, Dataset loading, Data preprocessing, Feature selection, CV scoring, CV shortlisting and Potential job applicants.

- In the above diagram, we follow the tasks such as uploading job applicants CV and then that application is loaded into dataset.
- We process the dataset by using machine learning techniques, after preprocessing the data we will extract the percentage based on the skills by using feature selection.
- Considering all the modules percentages we will analyze the CV of applicant and rate or rank their CV by minimum cut-off of all the modules.
- By using TF-IDF algorithm we will calculate and analyze the CV of job applicant.
- And this data is stored in database for further analysis and to calculate the percentage of data provided based on the skills.
- After scoring the CVs then we shortlist them for further process that is to communicate with the hr team for the interviews.

3.4. MODULES:

- ❖ Load cv
- ❖ View data
- ❖ Analyze one
- ❖ Analyze all

MODULES DESCRIPTION:

- Load CV: To analyze the cv at first we need to load the file containing the information about the person who want to analyze ,we load the any number of cv by just clicking the button add and it shows the popup window and we select the file using the path of the file by file chooser,1st main idea coming from this section in this section the company provides a basic email and they offer to all persons who are eligible for the job they need to send their resume to the particular email after completion of the time period the company head collect the all resumes and store those files anywhere in the system.
- View :This module enables the user or officer to view the content in the document and observe the format of the cv is suitable for your application and this module offers to head officials to read a particular cv and check for there is any tricks are apply for the grade.
- Analyze one :we can also analyze the one cv at a time by using the module analyze one panel , we simply type the file name in the text field provided in the same panel. After loading the file into the panel it analyze the cv and give the information about the candidate and his overall grade and overview in all fields.

- Analyze all: We can also analyze the all cv we loaded into the application in this windows shows the all the grades for the each applicant.
- Categorize: We can also find out the presongs based on their work category

3.5. FEASIBILITYSTUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential

I. ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

II. TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed

on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

III. SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.6. SYSTEM REQUIREMENTS

HARDWARE REQUIREMENTS:

- System : Intel core i5 processor.
- HardDisk : 1TB.
- Ram : 8GB.

SOFTWARE REQUIREMENTS:

- Operating Systems : Windows 10
- Coding Language : JAWASWINGS, AWT
- DataBase. : File System
- IDE : NETBEANS 15
- Documentation : MS Office

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1. INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OBJECTIVES

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that

all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

OUTPUTDESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

1. Convey information about past activities, current status or projections.
2. Future
3. Signal important events, opportunities, problems, or warnings.
4. Trigger an action

4.2. UMLDIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

- Complex applications need collaboration and planning from multiple teams and hence require a clear and concise way to communicate amongst them.
- Businessmen do not understand code. So UML becomes essential to communicate with non- programmers about essential requirements, functionalities, and processes of the system.
- A lot of time is saved down the line when teams can visualize processes, user interactions, and the static structure of the system.
- We use UML diagrams to portray the behaviour and structure of the system.
- UML helps software engineers, businessman, and system architects with modelling, design, analysis.

USE CASE DIAGRAM:

Diagram is a use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis.

Its purpose is to present a graphical overview functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases.

- Use-case diagrams are used in many situations.

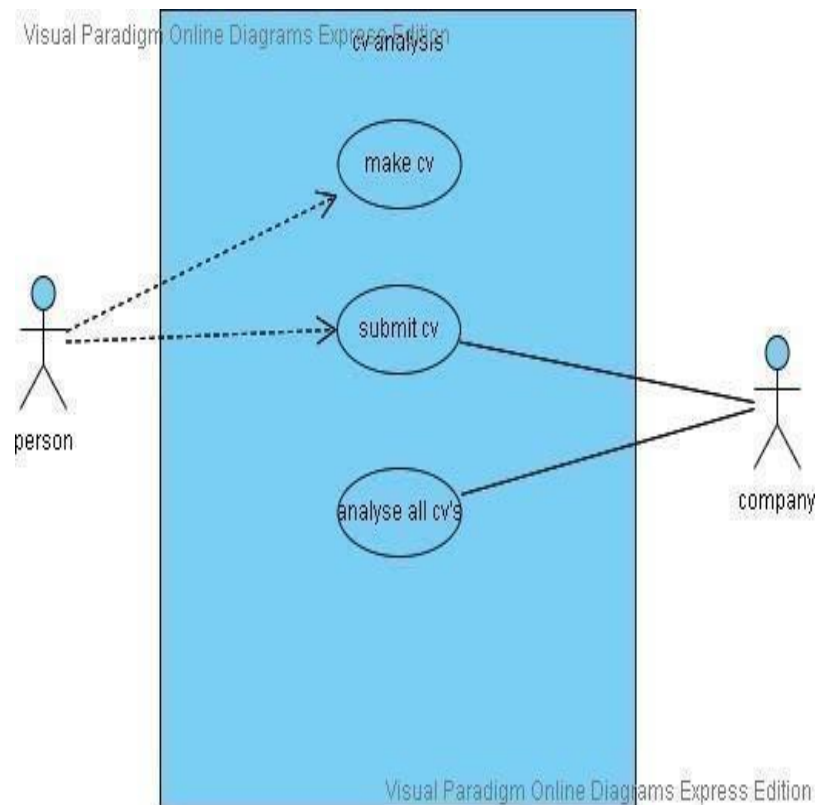


Fig 4.2.1 USE CASE DIAGRAM

- The above use case diagram describes about analyzing the CV of all the students.

- It describes a function that a system performs to achieve the user's goal.
- Students are represented as person and the manager is represented as a company.
- First a person designs their own CV and submits that CV to respective company for further process.
- Once the person submits the CV to the company, the manager analyses the CV and sends the status of their application to the person.
- UML diagrams are those which describes about the relation between two entities using entities and the use cases.
- Let us consider situations where we use use-case diagrams, before starting a project, we can create use-case diagrams to model a business so that all participants in the project share an understanding of the workers, customers, and activities of the business.
- During the analysis and design phases, we can use the use cases and actors from our use case diagrams to identify the classes that the system requires.

SEQUENCE DIAGRAM:

A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

- A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modelling Language(UML).
- among software objects in the Unified Modelling Language(UML).

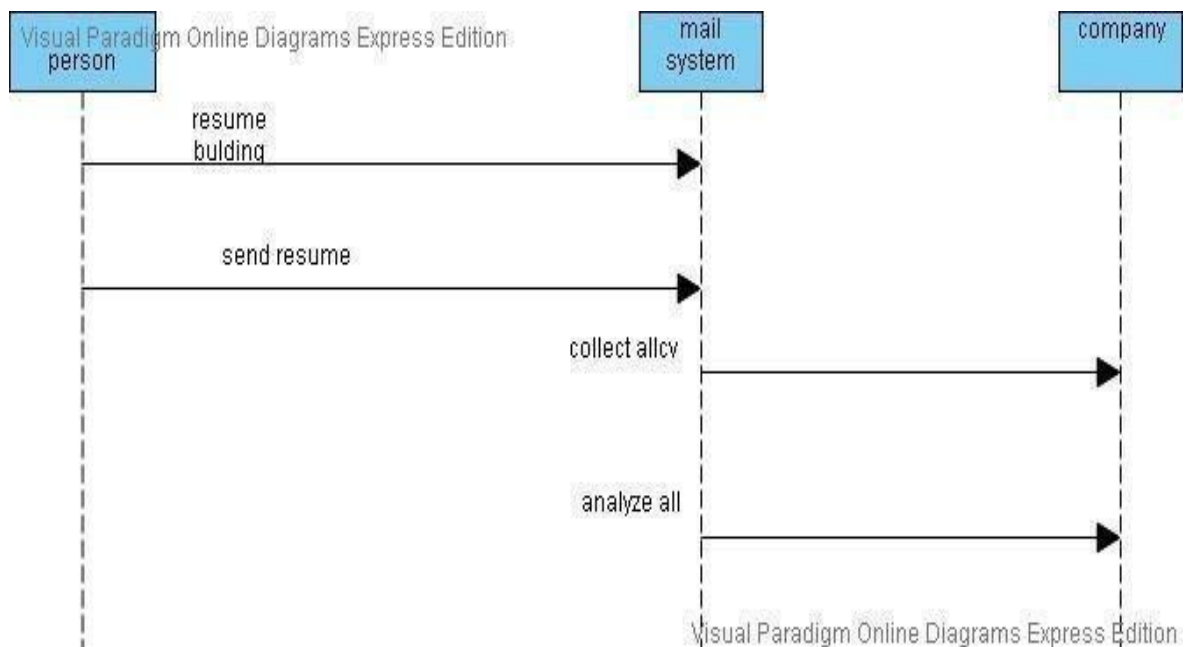


Fig 4.2.3 SEQUENCE DIAGRAM

ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

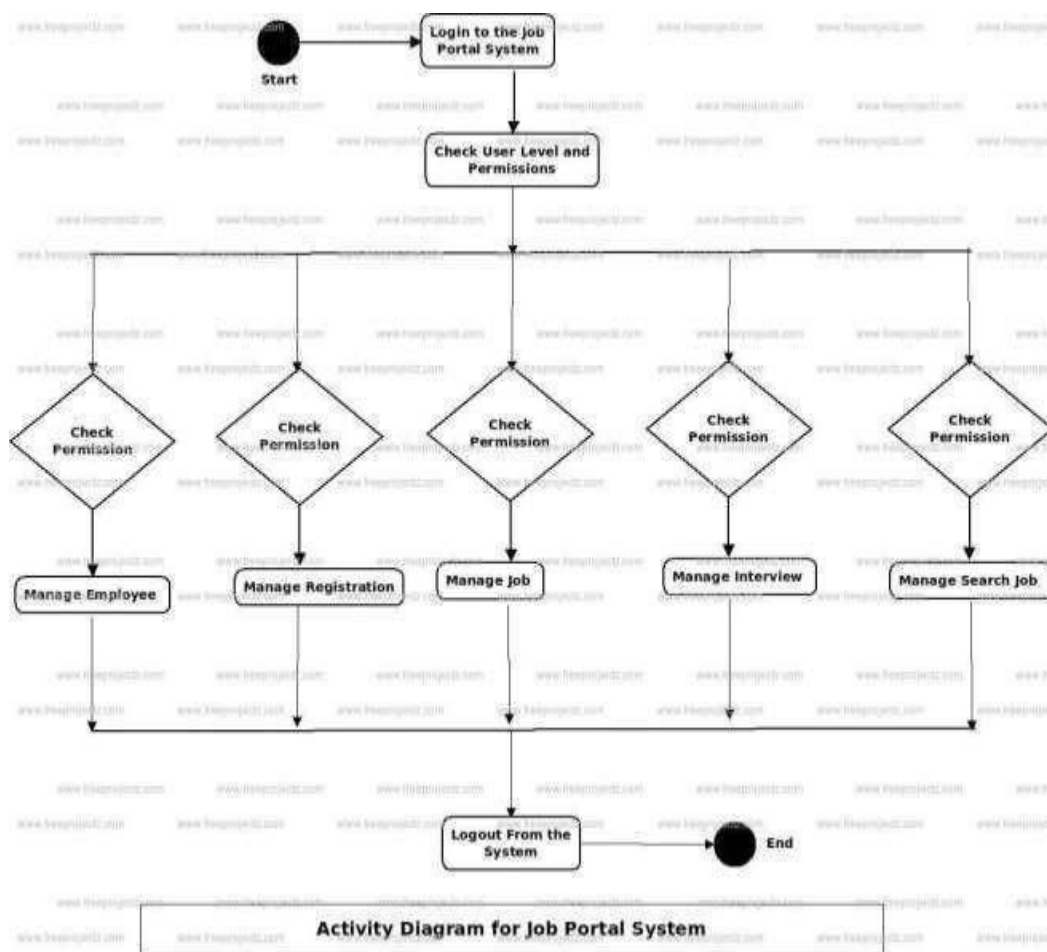


Fig 4.2.2 ACTIVITY DIAGRAM

IMPLEMENTATION

5. IMPLEMENTATION

5.1 SOFTWARE ENVIRONMENT:

5.1.1. Java Technology

Java technology is both a programming language and a platform.

1. Java programming language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

With most programming languages, you either compile or interpret a program so that you can run it on your computer. The Java programming language is unusual in that a program is both compiled and interpreted. With the compiler, first you translate a program into an intermediate language called Java byte codes —the platform-independent codes interpreted by the interpreter on the Java platform. The interpreter parses and runs each Java byte code instruction on the computer. Compilation happens just once; interpretation occurs each time the program is executed.

The following figure illustrates how this works.

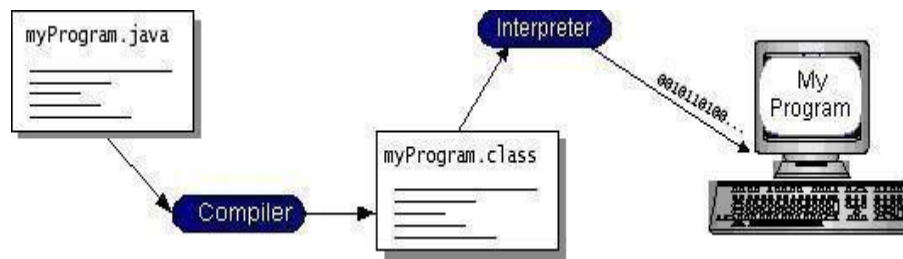


Fig 5.1.1 : java technology

You can think of Java byte codes as the machine code instructions for the Java Virtual Machine (Java VM). Every Java interpreter, whether it's a development tool or a Web browser that can run applets, is an implementation of the Java VM. Java byte codes help make “write once, run anywhere” possible. You can compile your program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the Java VM. That means that as long as a computer has a Java VM, the same program written in the Java programming language can run on Windows 2000, a Solaris workstation, or on an iMac.

The Java Platform

A platform is the hardware or software environment in which a program runs. We've already mentioned some of the most popular platforms like Windows 2000, Linux, Solaris, and MacOS. Most platforms can be described as a combination of the operating system and hardware. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

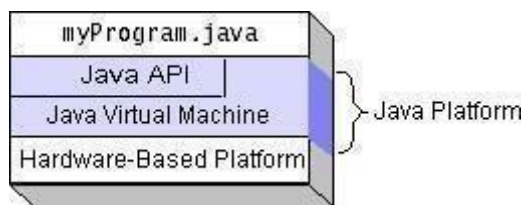
The Java platform has two components:

- . The Java Virtual Machine (JavaVM)
- . The Java Application Programming Interface (JavaAPI)

You've already been introduced to the Java VM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages. The next section, What Can JavaTechnology Do? Highlights what functionality some of the packages in the Java API provide.

The following figure depicts a program that's running on the Java platform. As the figure shows, the Java API and the virtual machine insulate the program from the hardware.



Native code is code that after you compile it, the compiled code runs on a specific hardware platform. As a platform-independent environment, the Java platform can be a bit slower than native code. However, smart compilers, well-tuned interpreters, and Just in time byte code compilers can bring performance close to that of native code without threatening portability.

5.1.2 Developing J2ME applications

Introduction In this section, we will go over some considerations you need to keep in mind when developing applications for smaller devices. We'll take a look at the way the compiler is invoked when using J2SE to compile J2ME applications. Finally, we'll explore packaging and deployment and the role pre-verification plays in this process.

5.1.3 : Design considerations for small devices

Developing applications for small devices requires you to keep certain strategies in mind during the design phase. It is best to strategically design an application for a small device before you begin coding. Correcting the code because you failed to consider all of the "gotchas" before developing the application can be a painful process. Here are some design strategies to consider:

- * Keep it simple. Remove unnecessary features, possibly making those features a separate, secondary application.
- * Smaller is better. This consideration should be a "no brainer" for all developers. Smaller applications use less memory on the device and require shorter installation times. Consider packaging your Java applications as compressed JavaArchive (jar)files.
- * Minimize run-time memory use. To minimize the amount of memory used at run time, use scalar types in place of object types. Also, do not depend on the garbage collector. You should manage the memory efficiently yourself by setting object references to null when you are finished with them. Another way to reduce run-time memory is to use lazy instantiation, only allocating objects on an as-needed basis. Other ways of reducing overall and peak memory use on small devices are to release resources quickly, reuse objects, and avoid exceptions.

Configurations overview

The configuration defines the basic run-time environment as a set of core classes and a specific JVM that run on specific types of devices. Currently, two configurations exist for J2ME, though others may be defined in the future:

Connected Limited Device Configuration (CLDC) is used specifically the KVM for 16-bit or 32-bit devices with limited amounts of memory. This is the configuration (and the virtual machine) used for developing small J2ME

applications. Its size limitations make CLDC more interesting and challenging (from a development point of view) than CDC. CLDC is also the configuration that we will use for developing our drawing tool application. An example of a small wireless device running small applications is a Palm hand-held computer.

Connected Device Configuration (CDC) is used with the C virtual machine (CVM) and is used for 32-bit architectures requiring more than 2 MB of memory. An example of such a device is a Net TV box.

What is a J2ME profile?

As we mentioned earlier in this tutorial, a profile defines the type of device supported. The Mobile Information Device Profile (MIDP), for example, defines classes for cellular phones. It adds domain-specific classes to the J2ME configuration to define uses for similar devices. Two profiles have been defined for J2ME and are built upon CLDC: KJava and MIDP. Both KJava and MIDP are associated with CLDC and smaller devices. Profiles are built on top of configurations. Because profiles are specific to the size of the device (amount of memory) on which an application runs, certain profiles are associated with certain configurations.

A skeleton profile upon which you can create your own profile, the Foundation Profile, is available for CDC.

Profile 1: KJava

KJava is Sun's proprietary profile and contains the KJava API. The KJava profile is built on top of the CLDC configuration. The KJava virtual machine, KVM, accepts the same byte codes and class file format as the classic J2SE virtual machine. KJava contains a Sun-specific API that runs on the Palm OS. The KJava API has a great deal in common with the J2SE Abstract Windowing Toolkit (AWT). However, because it is not a standard J2ME package, its main package is `com.sun.kjava`. We'll learn more about the KJava API later in this tutorial when we develop some sample applications.

Profile 2: MIDP

MIDP is geared toward mobile devices such as cellular phones and pagers. The MIDP, like KJava, is built upon CLDC and provides a standard run-time environment that allows new applications and services to be deployed dynamically on end user devices. MIDP is a common, industry standard profile for mobile devices that is not dependent on a specific vendor. It is a complete and supported foundation for mobile application development. MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP-specific packages.

- * java.lang

5.2. SOURCE CODE

```
import java.awt.Color;
import java.awt.Font;
import java.awt.GradientPaint;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.RenderingHints;
import java.awt.event.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import javax.swing.*;
import java.io.*;
import java.text.DecimalFormat;
import java.util.Arrays;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.border.BevelBorder;
import javax.swing.border.Border;
import javax.swing.filechooser.FileSystemView;
public class CVA implements ActionListener {
    String alllocation = "";
    int nfiles = 0;
    String shortlist = "";
    //log in
    JPanel login;
    JTextArea a;
    JPanel p;
    JFrame f;
    JLayeredPane lp;
    //head
    JPanel head;
    JButton add, viewone, analyzeone, analyzeall;
    //head
    //navigation
    JPanel navigation;
    String cvlist = "";
    JTextArea navtext;
    //navigation
    //body
```

```

JPanel view;
JPanel filenameone, dataone;
JLabel enterfilename; JButton
viewdateone;
JTextField getfile;
JTextArea onefiledata;
//analyze
JLabel l1, filenameone1, personname, grade;
JTextField getfile1;
JButton gradeview;
JPanel graderesultone1, graderesultone2;
JTextArea grade1data;
JProgressBar jb; int i = 0;
//analyze
//analyze all
JPanel all, alt;
JLabel a1name, algrade;
JTextArea alttext;
//analyze all
JScrollPane pb;
//body
JTextField userid;
JPasswordField password;
JLabel loginl;
JButton loginb;
//shorting
String st = "", ad = "", sa = "";
int sectorno = 0; JButton
filtering;
JLabel filterl;
JPanel filter1, filter2;
JMenu filtert;
JMenuItem st1, ad1, sa1;
JMenuBar category; JTextArea
filterd;
public CVA() throws FileNotFoundException, IOException {
    userid = new JTextField("USER ID :");
    userid.setForeground(Color.WHITE);
    userid.setBackground(Color.BLACK); userid.setBounds(400,
    400, 200, 30); userid.addFocusListener(new FocusListener() {
        @Override
        public void focusGained(FocusEvent e) {
            if (userid.getText().equals("USER ID :")) {

```

```

        userid.setText("");
        userid.setForeground(Color.WHITE);
    }
}
@Override
public void focusLost(FocusEvent e) {
    if (userid.getText().isEmpty()) {
        userid.setForeground(Color.WHITE); userid.setText("USER
        ID :");
    }
}
});
password = new JPasswordField("PASSWORD");
password.setForeground(Color.WHITE);
password.setBackground(Color.BLACK); password.setBounds(400,
450, 200, 30); password.addFocusListener(new FocusListener() {
    @Override
    public void focusGained(FocusEvent e) {
        if (password.getText().equals("PASSWORD")) {
            password.setText("");
            password.setForeground(Color.WHITE);
        }
    }
    @Override
    public void focusLost(FocusEvent e) {
        if (password.getText().isEmpty()) { password.setForeground(Color.WHITE);
            password.setText("PASSWORD");
        }
    }
});
loginb = new JButton("LOG IN"); loginb.setBounds(450,
500, 100, 30); loginb.setForeground(Color.WHITE);
loginb.setBackground(Color.BLACK);
ImageIcon logini = new ImageIcon("C:\\\\Major Project\\wallpaper.jpg");
loginl = new JLabel(); loginl.setIcon(logini);
loginl.setBackground(Color.decode("#8ae6a2"));
loginl.setBounds(0, 0, 1200, 700);
loginl.add(password); loginl.add(userid);
loginl.add(userid);
loginl.add(password);
loginl.add(loginb); login = new
JPanel(); login.setBounds(0, 0,
1200, 700); login.setLayout(null);
login.add(loginl);

```

```
//===== head=====

Border mb1b = BorderFactory.createBevelBorder(BevelBorder.RAISED,
Color.LIGHT_GRAY, Color.GREEN);
    add = new JButton("ADD");
add.setFont(new Font("sarif", Font.BOLD, 20));
add.setBackground(Color.decode("#8ae6a2"));
add.setForeground(Color.yellow); add.setBounds(0, 0,
100, 40);
add.setBorder(null);
add.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent me) {
        add.setBorder(mb1b); add.setBackground(Color.CYAN);
        add.setForeground(Color.BLUE);
    }
    @Override
    public void mouseExited(MouseEvent me) {
        add.setBorder(BorderFactory.createEmptyBorder());
        add.setBackground(Color.decode("#8ae6a2")); add.setForeground(Color.yellow);
    }
});

viewone = new JButton("VIEW"); viewone.setFont(new
Font("sarif", Font.BOLD, 20));
viewone.setBackground(Color.decode("#8ae6a2"));
viewone.setForeground(Color.yellow);
viewone.setBounds(110, 0, 200, 40);
viewone.setBorder(null);
viewone.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseEntered(MouseEvent me) {
        viewone.setBorder(mb1b); viewone.setBackground(Color.CYAN);
        viewone.setForeground(Color.BLUE);
    }
    @Override
    public void mouseExited(MouseEvent me) {
        viewone.setBorder(BorderFactory.createEmptyBorder());
        viewone.setBackground(Color.decode("#8ae6a2"));
        viewone.setForeground(Color.yellow);
    }
});

analyzeone = new JButton("ANALYZE_ONE");
analyzeone.setFont(new Font("sarif", Font.BOLD, 20));
analyzeone.setBackground(Color.decode("#8ae6a2"));
analyzeone.setForeground(Color.yellow);
```

```

analyzeone.setBounds(320, 0, 200, 40);
analyzeone.setBorder(null); analyzeone.addMouseListener(new
MouseListener() {
    @Override
    public void mouseEntered(MouseEvent me) { analyzeone.setBorder(mb1b);
        analyzeone.setBackground(Color.CYAN);
        analyzeone.setForeground(Color.BLUE);
    }
    @Override
    public void mouseExited(MouseEvent me) {
        analyzeone.setBorder(BorderFactory.createEmptyBorder());
        analyzeone.setBackground(Color.decode("#8ae6a2"));
        analyzeone.setForeground(Color.yellow);
    }
});
analyzeall = new JButton("ANALYZE_ALL");
analyzeall.setFont(new Font("sarif", Font.BOLD, 20));
analyzeall.setBackground(Color.decode("#8ae6a2"));
analyzeall.setForeground(Color.yellow); analyzeall.setBounds(530, 0,
200, 40);
analyzeall.setBorder(null); analyzeall.addMouseListener(new
MouseListener() {
    @Override
    public void mouseEntered(MouseEvent me) { analyzeall.setBorder(mb1b);
        analyzeall.setBackground(Color.CYAN);
        analyzeall.setForeground(Color.BLUE);
    }
    @Override
    public void mouseExited(MouseEvent me) {
        analyzeall.setBorder(BorderFactory.createEmptyBorder());
        analyzeall.setBackground(Color.decode("#8ae6a2"));
        analyzeall.setForeground(Color.yellow);
    }
});
filtering = new JButton("CATEGORIZE"); filtering.setFont(new
Font("sarif", Font.BOLD, 20));
filtering.setBackground(Color.decode("#8ae6a2"));
filtering.setForeground(Color.yellow); filtering.setBounds(740, 0,
200, 40);
filtering.setBorder(null); filtering.addMouseListener(new
MouseListener() {
    @Override
    public void mouseEntered(MouseEvent me) {
        filtering.setBorder(mb1b); filtering.setBackground(Color.CYAN);
        filtering.setForeground(Color.BLUE);
    }
});

```



```

    }
    @Override
    public void mouseExited(MouseEvent me) {
        filtering.setBorder(BorderFactory.createEmptyBorder());
        filtering.setBackground(Color.decode("#8ae6a2"));
        filtering.setForeground(Color.yellow);
    }
});
head = new JPanel(); head.setBounds(0, 0, 1200,
40);
head.setBackground(Color.decode("#8ae6a2"));
head.setLayout(null); head.add(add);
head.add(viewone);
head.add(analyzeone);
head.add(analyzeall);
head.add(filtering);
//===== head=====
//===== navigation=====
navtext = new JTextArea(); navtext.setBounds(0, 0,
200, 660);
navtext.setBackground(Color.decode("#663300"));
navtext.setForeground(Color.white);
navtext.setFont(new Font("sarif", Font.BOLD, 20));
navtext.setBorder(null); navtext.setText("\nCV LIST\n");
navtext.setVisible(true); navtext.setEditable(false);
navigation = new JPanel();
navigation.setBounds(0, 40, 200, 660);
navigation.setBackground(Color.BLACK);
navigation.setLayout(null);
navigation.add(navtext);
//===== navigation=====
//===== body=====
/*view = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY
        ); int w = getWidth(); int h =
        getHeight();
        Color color1 = Color.decode("#cc2b5e");
        Color color2 = Color.decode("#753a88");

```

```

        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);
        g2d.setPaint(gp); g2d.fillRect(0, 0, w, h);

    }
};*/

view = new JPanel(); view.setBounds(200,
40, 1000, 660);
view.setBackground(Color.CYAN);
view.setLayout(null);
//view
enterfilename = new JLabel("ENETR FILE NAME : ");
enterfilename.setBounds(10, 10, 200, 40); enterfilename.setFont(new
Font("sarif", Font.BOLD, 20));
enterfilename.setForeground(Color.PINK);
getfile = new JTextField(30); getfile.setBounds(220,
10, 300, 40); getfile.setFont(new Font("sarif",
Font.BOLD, 20));
viewdateone = new JButton("VIEW DATA");
viewdateone.setBounds(530, 10, 200, 40);
viewdateone.setFont(new Font("sarif", Font.BOLD, 20));
viewdateone.setForeground(Color.yellow);
viewdateone.setBackground(Color.decode("#76871e"));
viewdateone.setBorder(null); filenameone = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY
    ); int w = getWidth(); int h =
        getHeight();
        Color color1 = Color.decode("#cc205e");
        Color color2 = Color.decode("#754a88");
        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);
        g2d.setPaint(gp); g2d.fillRect(0, 0, w, h);
    }
};
filenameone.setBounds(200, 40, 1000, 60); filenameone.setBackground(Color.BLUE);
filenameone.setLayout(null); filenameone.add(getfile);
filenameone.add(viewdateone);
filenameone.add(enterfilename);
onefiledata = new JTextArea(); onefiledata.setLineWrap(true);
onefiledata.setWrapStyleWord(true);
onefiledata.setBackground(Color.decode("#0c9686"));
onefiledata.setForeground(Color.BLACK);

```

```

onefiledata.setBorder(null); pb = new
JScrollPane(onefiledata);
pb.setBounds(00, 00, 1000, 600);
//view
//analyze one personname = new JLabel("");
personname.setBounds(10, 10, 500, 40);
personname.setFont(new Font("sarif", Font.BOLD, 40));
personname.setForeground(Color.BLACK);
jb = new JProgressBar(0, 2000);
jb.setForeground(Color.red); jb.setBounds(510,
10, 200, 40);
jb.setValue(0); jb.setStringPainted(true);
jb.setVisible(false);
grade = new JLabel("");
grade.setBounds(810, 10, 200, 60); grade.setFont(new
Font("sarif", Font.BOLD, 40));
grade.setForeground(Color.PINK);
l1 = new JLabel("DATAANALYSIS LIKE BELOW");
l1.setBounds(20, 70, 600, 30); l1.setFont(new
Font("sarif", Font.BOLD, 30));
l1.setForeground(Color.magenta);
filenameone1 = new JLabel("ENETR FILE NAME : ");
filenameone1.setBounds(10, 10, 200, 40); filenameone1.setFont(new
Font("sarif", Font.BOLD, 20));
filenameone1.setForeground(Color.PINK);
getfile1 = new JTextField(30); getfile1.setBounds(220,
10, 300, 40); getfile1.setFont(new Font("sarif",
Font.BOLD, 20));
gradeview = new JButton("ANALYZE");
gradeview.setBounds(540, 10, 200, 40);
gradeview.setFont(new Font("sarif", Font.BOLD, 20));
gradeview.setForeground(Color.yellow);
gradeview.setBackground(Color.decode("#76871e"));
gradeview.setBorder(null); graderesultone1 = new
JPanel() {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALIT
Y); int w = getWidth(); int h =
getHeight();
        Color color1 = Color.decode("#cc4b5e");
        Color color2 = Color.decode("#752a88");

```

```

        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);
        g2d.setPaint(gp); g2d.fillRect(0, 0, w, h);
    }
};

graderesultone1.setBounds(200, 40, 1000, 60);
graderesultone1.setBackground(Color.BLUE);
graderesultone1.setLayout(null); graderesultone1.add(getfile1);
graderesultone1.add(gradeview);
graderesultone1.add(filenameone1);

grade1dtata = new JTextArea(); grade1dtata.setBounds(0,
100, 1000, 500);
grade1dtata.setBackground(Color.decode("#d7de64"));
grade1dtata.setForeground(Color.decode("#d4af37"));
grade1dtata.setFont(new Font("sarif", Font.BOLD, 25));
grade1dtata.setBorder(null); graderesultone2 = new JPanel();
graderesultone2.setBounds(200, 100, 1000, 600);
graderesultone2.setBackground(Color.decode("#d7de64"));
graderesultone2.setForeground(Color.decode("#d4af37"));
graderesultone2.setLayout(null);
graderesultone2.add(personname);
graderesultone2.add(grade); graderesultone2.add(grade1dtata);
graderesultone2.add(jb);

//analyze one
//analyze all

alname = new JLabel("NAME OF THE APPLICANT");
alname.setBounds(10, 10, 600, 40); alname.setFont(new
Font("sarif", Font.BOLD, 40));
alname.setForeground(Color.PINK);
algrade = new JLabel("GRADE");
algrade.setBounds(710, 10, 200, 40);
algrade.setFont(new Font("sarif", Font.BOLD, 40));
algrade.setForeground(Color.PINK); all = new JPanel()
{
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALIT
Y); int w = getWidth(); int h =
getHeight();
        Color color1 = Color.decode("#cc4b5e");
        Color color2 = Color.decode("#752a88");
        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);
        g2d.setPaint(gp); g2d.fillRect(0, 0, w, h);
    }
};

```

```

    }
};
all.setBounds(200, 40, 1000, 60); all.setBackground(Color.BLUE);
all.setLayout(null);
all.add(alname); all.add(algrade);
altext = new JTextArea(); altext.setBounds(0, 100,
1000, 500);
altext.setBackground(Color.decode("#1b1e23"));
altext.setForeground(Color.decode("#d4af37"));
altext.setFont(new Font("sarif", Font.BOLD, 25));
altext.setBorder(null); alt = new JPanel();
alt.setBounds(200, 00, 1000, 700);
alt.setBackground(Color.decode("#1b1e23"));
alt.setForeground(Color.decode("#d4af37"));
alt.setLayout(null);
alt.add(altext);
//
// filtering
filterl = new JLabel("SELECT THE CATEGORY ");
filterl.setBounds(10, 10, 600, 40); filterl.setFont(new
Font("sarif", Font.BOLD, 40));
filterl.setForeground(Color.PINK);
st1 = new JMenuItem("Software Tester");
st1.setBackground(Color.yellow); ad1 = new
JMenuItem("Application developer");
ad1.setBackground(Color.yellow); sa1 = new
JMenuItem("Software analyst ");
sa1.setBackground(Color.yellow);
filtert = new JMenu("          select          ");
filtert.setForeground(Color.WHITE); filtert.setBackground(Color.BLACK);
filtert.setBorder(BorderFactory.createLineBorder(Color.CYAN, 5));
filtert.add(st1); filtert.add(ad1); filtert.add(sa1);
category = new JMenuBar(); category.setForeground(Color.WHITE);
category.setBackground(Color.BLACK);
category.add(filtert); category.setBounds(650,
5, 150, 40); filter1 = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY
        ); int w = getWidth(); int h =
        getHeight();
        Color color1 = Color.decode("#cc4b5e");

```

```

        Color color2 = Color.decode("#752a88");
        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);
        g2d.setPaint(gp); g2d.fillRect(0, 0, w, h);
    }
};

    filter1.setBounds(200, 40, 1000, 60);
filter1.setBackground(Color.BLUE);
filter1.setLayout(null); filter1.add(filter1);
filter1.add(category);
filterd = new JTextArea(); filterd.setBounds(0, 100,
1000, 500);
filterd.setBackground(Color.decode("#1b1e23"));
filterd.setForeground(Color.decode("#d4af37"));
filterd.setFont(new Font("sarif", Font.BOLD, 25));
filterd.setBorder(null);
filter2 = new JPanel(); filter2.setBounds(200, 00,
1000, 700);
filter2.setBackground(Color.decode("#1b1e23"));
filter2.setForeground(Color.decode("#d4af37"));
filter2.setLayout(null); filter2.add(filterd);
//
dataone = new JPanel() {
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        Graphics2D g2d = (Graphics2D) g;
        g2d.setRenderingHint(RenderingHints.KEY_RENDERING,
RenderingHints.VALUE_RENDER_QUALITY
        ); int w = getWidth(); int h =
        getHeight();
        Color color1 = Color.decode("#cb2b5e");
        Color color2 = Color.decode("#783a88");
        GradientPaint gp = new GradientPaint(0, 0, color1, 0, h, color2);
        g2d.setPaint(gp); g2d.fillRect(0, 0, w, h);
    }
};;
dataone.setBounds(200, 100, 1000, 600);
dataone.setLayout(null);
dataone.add(pb);
//===== body=====
//===== all layers===== lp
= new JLayeredPane(); lp.setBounds(0, 0, 1200, 700);
lp.setLayout(null); lp.setBackground(Color.GRAY);
lp.add(head); lp.add(navigation);

```

```

lp.add(view);
//===== all layers=====
f= new JFrame("CV SORT LISTING APPLICATION");
f.setSize(1200, 700);
f.setLayout(null);
f.setVisible(true);
f.add(login); //head
add.addActionListener(this);
viewone.addActionListener(this);
analyzeone.addActionListener(this);
analyzeall.addActionListener(this);
//head //body
viewdateone.addActionListener(this);
//body //analyze
one
gradeview.addActionListener(this);
//
//login loginb.addActionListener(this);
filtering.addActionListener(this);
st1.addActionListener(this);
ad1.addActionListener(this);
sa1.addActionListener(this);
} public static void main(String[] args) throws IOException {

CVA c = new CVA(); }

```

@Override

```

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == loginb) { String
        userid1;
        char[] correctpassword = {'M', 'a', 'j', 'o', 'r'};
        userid1 = userid.getText(); char[] psw =
        password.getPassword();
        if (userid1.equals("Batch14") && Arrays.equals(correctpassword, psw)) {
            f.remove(login);
            f.add(lp);
            f.repaint();
        } else {
            userid.setText(" "); password.setText("
");
        }
    } if (e.getSource() == add) {
        lp.removeAll(); shortlist =
        ""; filterd.setText("");
        lp.add(head);
    }
}

```

```

        lp.add(navigation);
        lp.add(view);
        altext.setText("");
        personname.setText("");
        grade.setText("");
        jb.setVisible(false);
        jb.setValue(0); i =
        0;

        getfile.setText(""); onefiledata.setText("");
        getfile1.setText(""); grade1data.setText("");

        JFileChooser jfc = new
JFileChooser(FileSystemView.getFileSystemView().getHomeDirectory()
        ); int returnValue = jfc.showOpenDialog(null); // int returnValue =
        jfc.showSaveDialog(null);
        if (returnValue == JFileChooser.APPROVE_OPTION) {
            File selectedFile = jfc.getSelectedFile();
            cvlist += selectedFile; String s1 =
            navtext.getText(); navtext.setText(s1 + "\n"
            + selectedFile.getName()); alllocation +=
            "==">" + selectedFile.getName() + "==">" +
            selectedFile.getAbsolutePath(); nfiles++;
        }
    }
    if (e.getSource() == viewone) {
        shortlist = "";
        filterd.setText("");
        getfile.setText("");
        altext.setText(""); shortlist =
        ""; onefiledata.setText("");
        personname.setText("");
        grade.setText("");
        jb.setVisible(false);
        jb.setValue(0); i =
        0;

        getfile1.setText("");
        grade1data.setText("");
        lp.removeAll(); lp.add(head);
        lp.add(navigation);
        lp.add(filenameone);
        lp.add(dataone);
    }
    if (e.getSource() == viewdateone) {
        shortlist = ""; filterd.setText("");
        String separatepath[] = alllocation.split("==">");
        altext.setText("");
        shortlist = ""; String

```



```

name = ""; String line
= "";
for (int i = 1; i < separatepath.length; i = i + 2) {
    String ss = getfile.getText(); if
    (ss.equals(separatepath[i])) {
        try {
            File file = new File(separatepath[i + 1]);
            BufferedReader bf = null; try {
bf = new BufferedReader(new FileReader(file));
            } catch (FileNotFoundException ex) {
                Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex); }
            line = bf.readLine(); while
            (line != null) {
                name += line + "\n"; line =
                bf.readLine();
            }
            } catch (IOException ex) {
                Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex); }
        }
    } onfiledata.setText(name);
}
if (e.getSource() == analyzeone) {
    shortlist = ""; filterd.setText("");
    altext.setText(""); getfile.setText("");
    onfiledata.setText("");
    personname.setText("");
    grade.setText(""); jb.setVisible(false);
    jb.setValue(0); i =
    0;
    getfile1.setText("");
    grade1data.setText("");
    lp.removeAll(); lp.add(head);
    lp.add(navigation);
    lp.add(gradesresultone1);
    lp.add(gradesresultone2);
}
if (e.getSource() == analyzeall) {
    shortlist = ""; filterd.setText("");
    ad = ""; sa =
    "";
    st = "";
    altext.setText("."); getfile.setText("");
    onfiledata.setText("");

```

```

personname.setText("");
grade.setText("");
f.repaint();
String separatepath[] = allocation.split("==>"); //
for(int j=1;j<=nfiles;j++){ f.repaint();
String separatepath1[] = allocation.split("==>");
String resultarray[] = new String[separatepath1.length];
int c = 0; for (int i = 1; i < separatepath1.length; i = i + 2)
{
    String name1 = ""; String line
    = ""; double totalpercentage =
    0;
    try {
        File file = new File(separatepath1[i + 1]);
        BufferedReader bf = null; try {
            bf = new BufferedReader(new FileReader(file));
        } catch (FileNotFoundException ex) {
            Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);
        }
        line = bf.readLine(); while
        (line != null) {
            name1 += line + "\n"; line =
            bf.readLine();
        }
    } catch (IOException ex) {
        Logger.getLogger(CVA.class.getName()).log(Level.SEVERE, null, ex);
    }
    String split[] = name1.split("==>");
    String name = split[1];
    String skills = split[3];
    String sk[] = skills.split("=>");
    double skillsno = sk.length - 1;
    double k = skillsno / 8; double
    skpercent = k * 100;
    String ski = "skills percentage =" + skpercent;
    String experiance = split[4];
    String exp[] = experiance.split("\n");
    String experi = exp[2]; String
    qualifin = "";
    if (experi.contains("Application developer")) {
        qualifin = "Application developer"; sectorno =
        1;
    } else if (experi.contains("Software analyst")) {
        qualifin = "Software analyst"; sectorno = 2;
    }
}

```

```

    } else if(experi.contains("Software Tester")) {
        qualifin = "Software Tester"; sectorno = 3;
    } String exnumber = ""; for (char ch :
experi.toCharArray()) { if
(Character.isDigit(ch)) {
        exnumber = Character.toString(ch);
    }
    } double ex = Double.parseDouble(exnumber);
    double expercent = ex / 4 * 100;
String exi = "experience percentage =" + expercent;
    String degree = split[5];
    String ed[] = degree.split("\n");
    String de = ed[3]; String degp = ""; int
    coun = 1; for (char ch :
degree.toCharArray()) {
        if (Character.isDigit(ch)) { if
        (coun <= 2) {
            degp += Character.toString(ch);
            coun++;
        }
    }
}
}

```

5.3. OUTPUT SCREENS

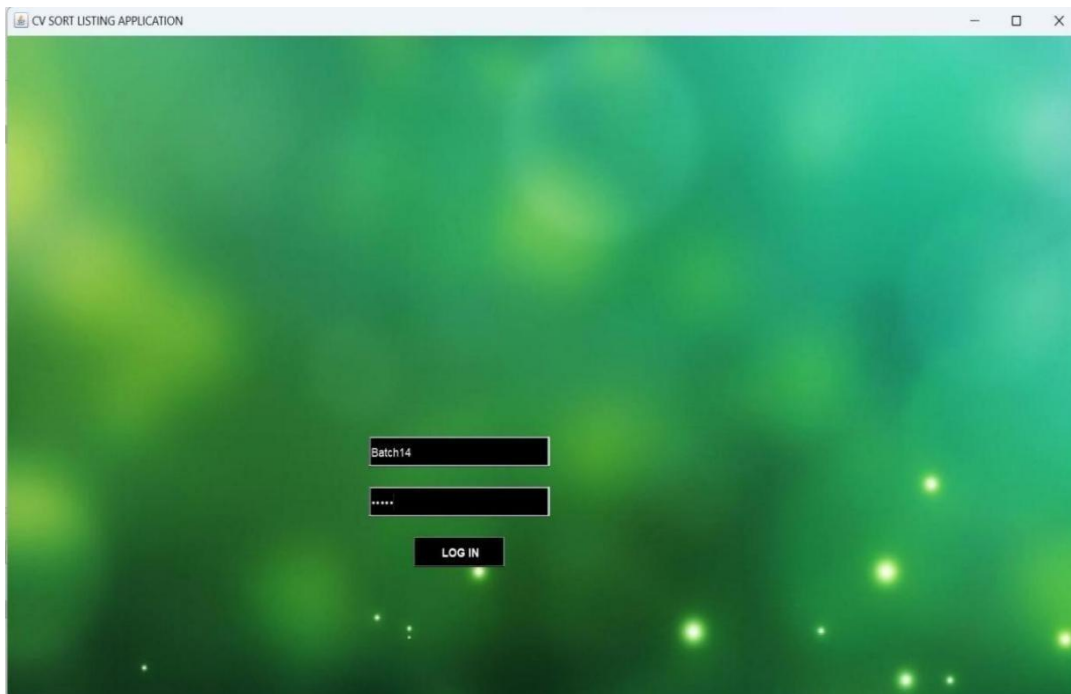


Fig 5.3.1: ADMIN LOGIN

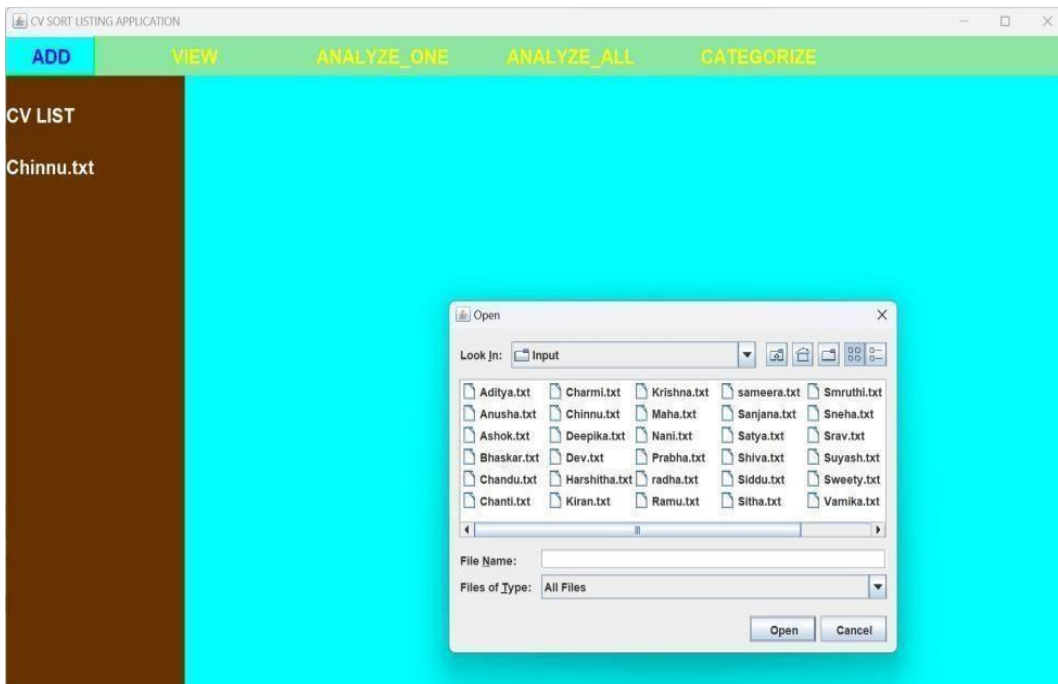


Fig 5.3.2:FILECHOOSE

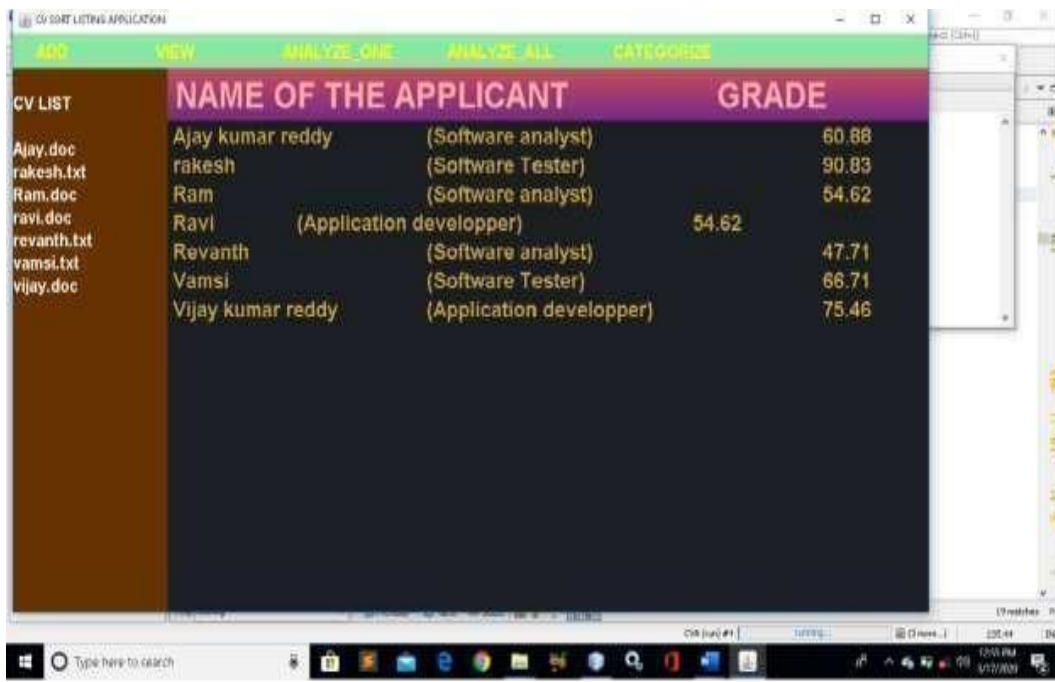


Fig 5.3.3 : readcontent

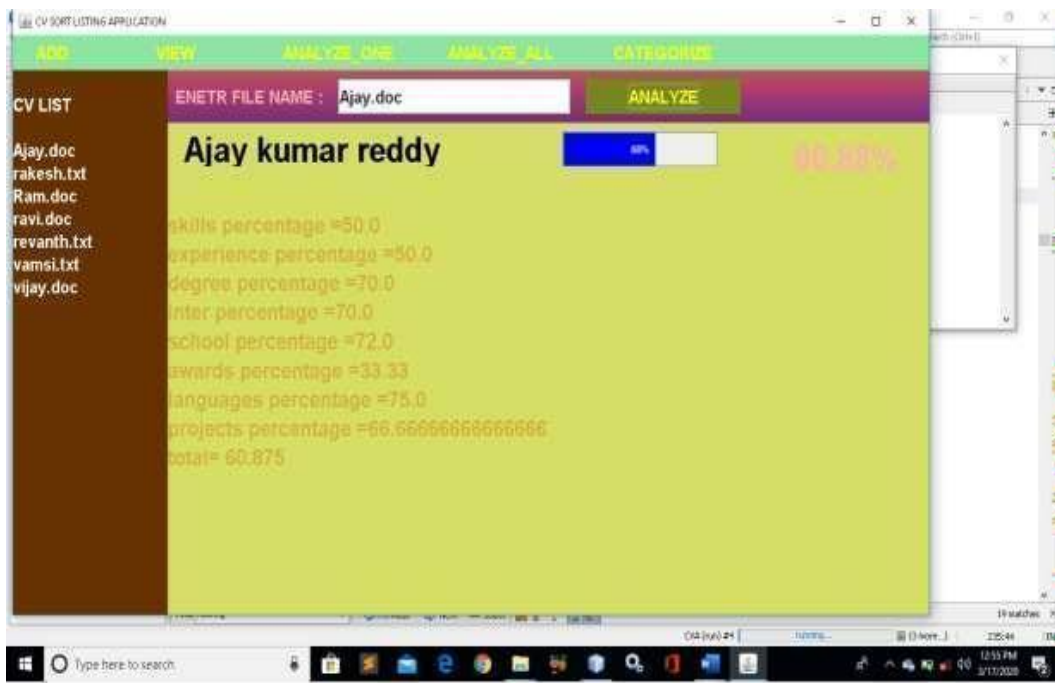


Fig 5.3.4 : analyze onecv

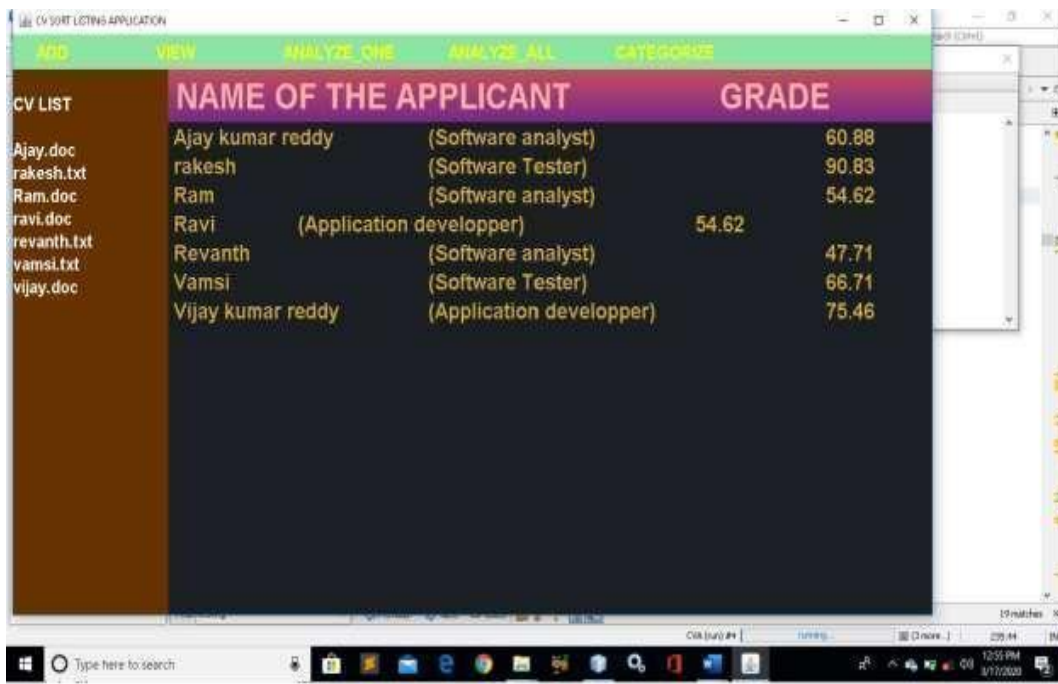


Fig 5.3.5: analyzeall

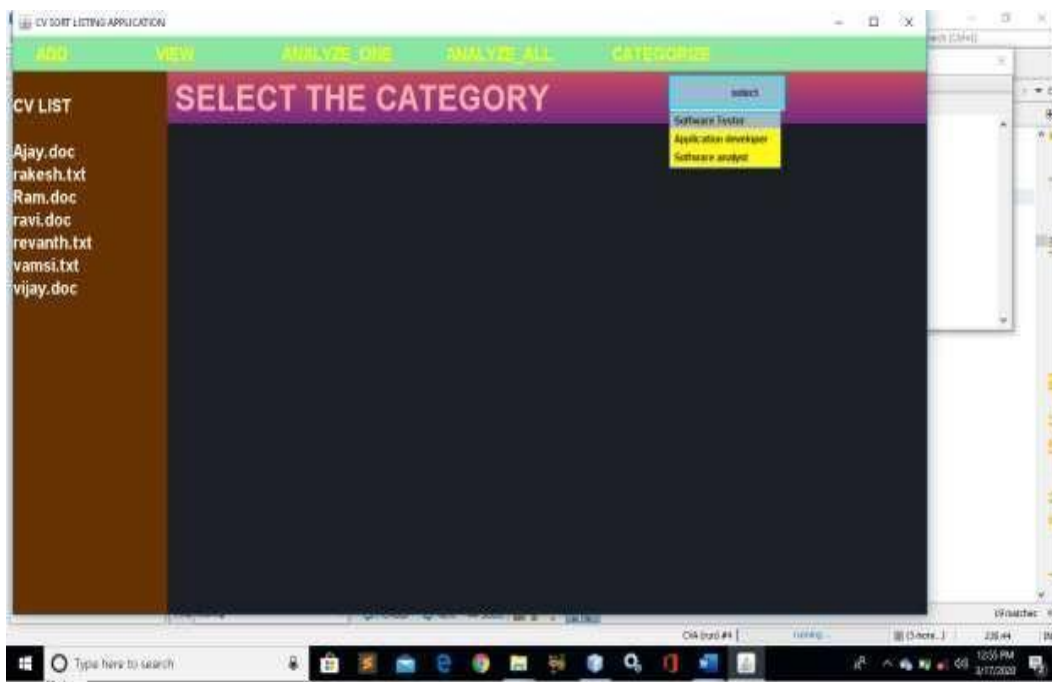


Fig5.3.6:categoriz

TYPES OF TESTING

6. TYPES OF TESTING

6.1. TEST STRATEGIES

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, sub-assemblies, assemblies and/or a finished product. 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 test. Each test type addresses a specific testing requirement.

TYPES OF TESTING

Unit testing: Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing:

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfactory, as

shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It issued to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

6.2. TEST OBJECTIVES:

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g.

components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

TEST CASES:

Test Case 1:

Test Case Description	Value
Test with a CV containing relevant experience	CV with relevant work experience and skills
Input	CV with relevant work experience and skills
Expected output	Predicted profile matches expected profile
Status	Passed

Test Case 2:

Test Case Description	Value
Test with a CV containing irrelevant experience	CV with irrelevant work experience
Input	CV with irrelevant work experience
Expected Output	Predicted profile differs from expected profile
Status	Failed

Test Case 3:

Test Case Description	Value
Test with a CV containing missing information	CV missing crucial sections or information
Input	CV missing crucial sections or information
Expected Output	Model handles missing data gracefully
Status	Passed

Test Case 4:

Test Case Description	Value
Test with a CV in different formats	CV in various formats (e.g., PDF, DOC, TXT)
Input	CV in various formats (e.g., PDF, DOC, TXT)
Expected Output	Model accepts and processes different formats
Status	Passed

Test Case 5:

Test Case Description	Value
Test with a CV from different industries	CV from different industries or job sectors
Input	CV from different industries or job sectors
Expected Output	Model generalizes well across different domains
Status	Passed

CONCLUSION

7.CONCLUSION

In recent years, all industries and IT companies feel high burden to analyze all resumes provided by the candidates who are willing to participate job requirement processes so we need automated system for analyzing and shortlisting the all resumes at a time and also calculate the grades of the persons and their individual grades in each section like experience ,skills, education background , languages known and number of projects did. So this project will help almost all companies who are willing to analyze all resumes at a time without putting any human effect. We can make our work easy to analyse and to shortlist the students in any specific category required in accordance of time. We have presented in this that the prediction of human personality by using standard questionnaires that is provided by the HR Department according to the job selection criteria. Candidates fill an online Curriculum Vitae(CV) which can be later on viewed by the Admin.

Candidates are provided with separate set keys for attempting the aptitude and personality based tests. CV analysis is performed using the CV filled by the candidate in the website. A machine learning approach has been used in analysis of data through content and collaborative filtering. Further, the test scores help in deciding the qualities in the candidates. Thus, the Curriculum Vitae(CV) is shortlisted for the recruitment process and a fair and appropriate decision is made by HR Department. Also data visualization model determines the overall performance of the students based on various factors. This analysis helps the Admin department to calculate the proficiency of candidates accurately.

FUTURE ENHANCEMENT

8. FUTURE ENHANCEMENT

In future cv analysis is required for each and every organization and adding additional features like “formats of the resumes are independent for analysis and applying for a job “.This will help to reduce the time taken to analyses all the cv that are received from many candidates who are willing to do the particular job that a company offers. which can predict the traits of an individual using the group of tweets posted by him. It is different from the approaches reviewed in the literature survey, in that it works with the group of tweets and does not take user’s profile into account. Further, the system makes use of the Hadoop framework to predict personality traits of multiple individuals at the same time.

Further, we can modify the existing system to perform sentiment analysis of social media data. Many more classification algorithms of machine learning can be integrated to provide much better functionalities. And this will help to reduce the time taken to analysis of all the cv that are received from all the candidates who are willing to do the particular job that a company offers.

Further, the efficiency and performance of the application can be tested and analysed. As, if the latest technologies arrived and habituated in the IT industries then we have to enhance the skills in addition to the skills. Based on that arrived technologies and whom are familiar or proficient in them can improve their percentage and also increases the weightage of the cv of the candidate. Finally, by calculating the percentage of the all the sections we can analyse the CV. This improves the efficiency and performance of the candidate.

BIBLIOGRAPHY

9. BIBLIOGRAPHY

1. **Anderson, N. H., & Barrios, A.A.** (1961). Primacy effects in personality impressionformation. *Journal of Abnormal and Social Psychology*, 63, 346-350.
<https://doi.org/10.1037/h0046719>
2. **Arnulf, J. K., Tegner, L., & Larssen, Ø.** (2010). Impression making by resume layout: Its impact on the probability of being shortlisted. *European Journal of Work and OrganizationalPsychology*, 19, 221-230.
<https://doi.org/10.1080/13594320902903613>
3. **Augustin, H. M.** (1991). The written job search: A comparison of the traditional and a nontraditional approach. *Business Communication Quarterly*, 54, 13-14.
<https://doi.org/10.1177/108056999105400304>
4. **Bendick, M., Jackson, C. W., & Romero, J. H.** (1996). Employment discrimination againstolder workers: An experimental study of hiring practices. *Journal of Aging & Social Policy*,8, 25-46.
https://doi.org/10.1300/J031v08n04_03
5. **Bertrand, M., & Mullainathan, S.** (2004). Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. *The AmericanEconomic Review*, 94, 991-1013.
<https://doi.org/10.1257/0002828042002561>
6. **Bird, C. P., & Puglisi, D. D.** (1986). Method of resume reproduction and evaluations of employment suitability. *The journal of business communication*, 23,31-39.
<https://doi.org/10.1177/002194368602300303>
7. **Blackburn-Brockman, E., & Belanger, K.** (2001). One page or two?: A national study of CPA recruiters' preferences for resume length. *The Journal of Business Communication*, 38, 29-57.
<https://doi.org/10.1177/002194360103800104>
8. **Bright, J. E. H., & Hutton, S.** (2000). The impact of competency statements on resumes for shortlisting decisions. *International Journal of Selection and Assessment*, 8, 41-53.
<https://doi.org/10.1111/1468-2389.00132>

9. **Brown, B. K., & Campion, M. A.** (1994). Biodata phenomenology: Recruiters' perceptions and use of biographical information in resume screening. *Journal of Applied Psychology*, 79, 897-908.

<https://doi.org/10.1037/0021-9010.79.6.897>

10. **Burns, G. N., Christiansen, N. D., Morris, M. B., Periard, D. A., & Coaster, J. A.** (2014). Effects of applicant personality on resume evaluations. *Journal of Business and Psychology*, 29, 573-591.

<https://doi.org/10.1007/s10869-014-9349-6>

11. **Campion, M. A.** (1978). Identification of variables most influential in determining interviewers' evaluations of applicants in a college placement center. *Psychological Reports*, 42, 947-952.

<https://doi.org/10.2466/pr0.1978.42.3.947>

12. **Catano, V. M., Wiesner, W. H., & Hackett, R. D.** (2016). *Recruitment and selection in Canada* (6th ed.). Toronto, ON: Nelson Education Ltd.

13. **Charney, D. H., & Rayman, J. R.** (1989). The role of writing quality in effective student resumes. *Journal of Business and Technical Communication*, 3, 36-53.

<https://doi.org/10.1177/105065198900300102>

14. **Charney, D. H., Rayman, J., & Ferreira-Buckley, L.** (1992). How writing quality influences readers' judgments of resumes in business and engineering. *Journal of Business and Technical Communication*, 6, 38-74.

<https://doi.org/10.1177/1050651992006001002>

15. **Chen, C., Huang, Y., & Lee, M.** (2011). Test of a model linking applicant resume information and hiring recommendations. *International Journal of Selection and Assessment*, 19, 374-387.

<https://doi.org/10.1111/j.1468-2389.2011.00566.x>

CURRICULUM VITAE ANALYSIS USING MACHINE LEARNING ALGORITHM

(Under the guidance of **Mrs.J . Lakshmi**, Associate Professor, Department
of Computer Science and Engineering,Tirumala Engineering College)

M.Anuhya

20NE1A05A1

CSE Department

anuhyamoram@gmail.com

Tirumala Engineering College

K.Gayathri Venkata Aasritha

20NE1A0571

CSE Department

kanamarlapudiaasritha@gmail.com

Tirumala Engineering College

N.Chandu Venkat Sai

20NE1A05b3

CSE Department

narnechandu5b3@gmail.com

Tirumala Engineering College

M.Dinesh Babu

20NE1A0597

CSE Department

mdineshbabu2002@gmail.com

Tirumala Engineering College

M.Harshitha

20NE1A0595

CSE Department

mannavaharshitha31@gmail.com

Tirumala Engineering College

Abstract— We describe a system called CV analysis machine, that processes a set of given free-form textual resumes (in English for IT domain), creates a standardized profile for each candidate, and for a given job description, identifies a ranked shortlist of k candidates, along with a matching score for each. The resume scoring function is hand-crafted, hierarchical, and uses domain-knowledge from recruitment experts. We describe a simple neural- network system that automatically learns some weights used in the scoring function, based on feedback about whether the candidate has their grades on each portion and calculate the overall grade of each candidate and find all grades at a time. Companies often receive thousands of resumes for each job posting and employ dedicated screeners to short list qualified applicants. In this paper, we present PROSPECT, a decision support tool to help these screeners shortlist resumes efficiently. Prospect mines resumes to extract salient aspects of candidate profiles like skills, experience in each skill, education details and experience. Extracted information is presented in the form of facets to aid recruiters in the task of screening. We also employ Information Retrieval techniques to rank all applicants for a given job opening. In our experiments we show that extracted information improves our ranking by 30% there by making screening tasks simpler and more efficient.

Keywords--e-recruitment,aptitude,personality prediction,credentials,Big five model of Personality

I.INTRODUCTION

Personality is the most important factor which reflects an individual, which keeps on varying . Tackling them is a tedious task for which we have implemented an approach to identify the personality and also provide with the recommendation. This will enable a more effective way to shortlist submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy, which can be legally justified. System will rank the experience and key skills required for a particular job position. The system will rank the CV ' s based on the experience and other key skills which are required for a particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy. This system will focus not only on qualification and in experience but also focuses on other important aspects, which are required for a particular job position. This system will help the human resource department to select the right candidate for a particular job profile, which in turn will provide an expert workforce for the organization.

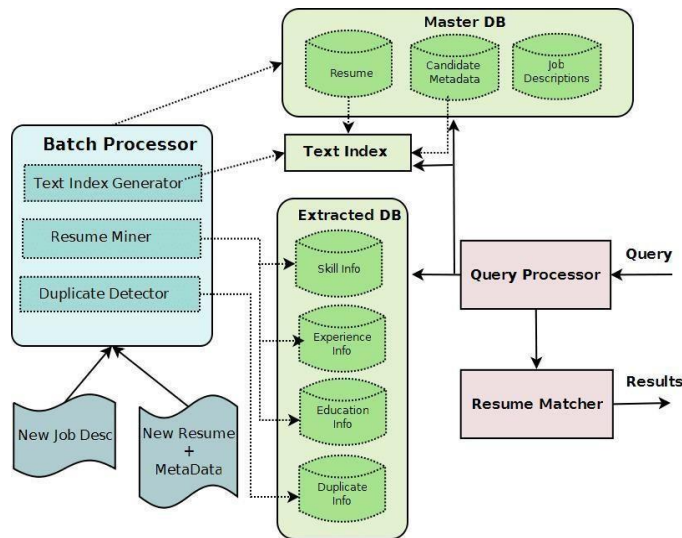
II.RELATED WORK

In this section, we present the related work and literature review of various techniques and algorithms used for the online selection process.

Big Five Personality Model (also known as Five Factor Model) has been used to predict the personality of the candidate which includes Openness, Conscientiousness, Extraversion , Agreeableness, and Neuroticism. For classifying the person Automated Personality Classification is used, which is used to classify the person from a large number of people.

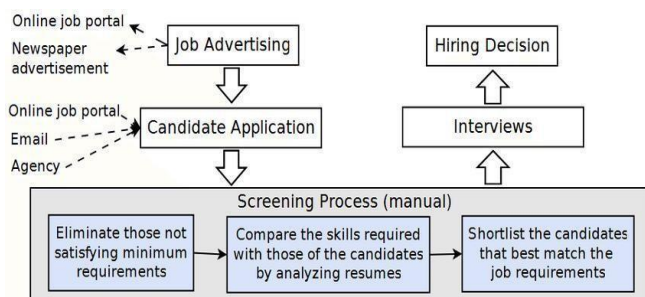
Recommendations using machine learning techniques have been used for the analysis of the CV. In literature, various evaluation tools have been used. One of the approaches has been mentioned which use a tool called “ PROSPECT ” for screening candidates for the recruitment.

candidates who were eligible for the job. The rank of each candidate acts as a score of how well the candidate's profile meets the specifications of the recruiters as well as cumulative score of the aptitude test. The recruiter could also analyze the p Personality of the candidate based on the result of the personality test. So, based on CV, aptitude test and the personality test the candidate would be selected.



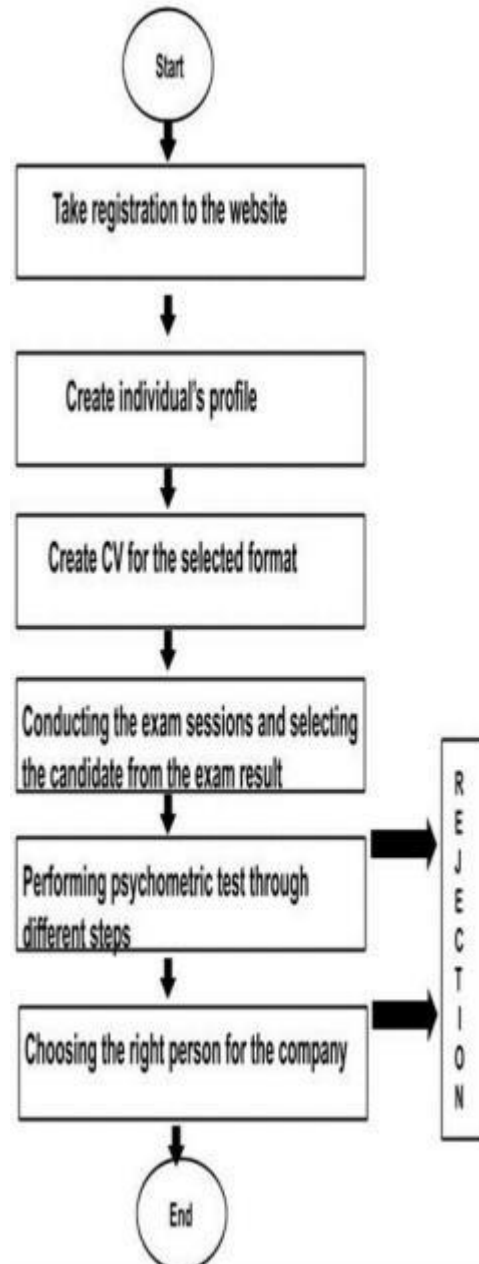
III. EXISTING SYSTEM

The candidates will register themselves with the required details in the CV form and upload the documents for verifying the authenticity of the information provided in the CV registration form. The candidates can then view the requirements and details of the job as specified by the admin or the recruiter. In order to apply for a job the candidate needs to be eligible for that job by satisfying all the requirements as stated by the recruiter. If a candidate applies for a specific job, the system checks whether the candidate meets all the requirements/parameters as specified by the recruiter. If the candidate satisfies the requirements/parameters, then the candidate's request for that job would be accepted else the system denies the candidate's request for that job.



Candidates can also give an online test, which will be conducted on personality questions as well as aptitude questions. After completing the online test, candidate can view their own test results in graphical representation with marks. The system would then derive and rank the

IV. PROPOSED SYSTEM

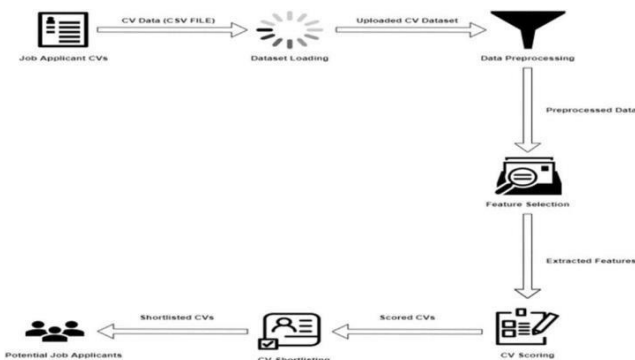


There is a huge workload on the human resource department to select the right candidate for a particular job profile which in turn would provide expert

workforce for the organization from a large pool of candidates. The proposed system will enable a more effective way to short list submitted candidate CVs from a large number of applicants providing a consistent and fair CV ranking policy. This can be legally justified. System will rank the experience and key skills required for a particular job position then system will rank the CV's based on the experience and other key skills which are required for a particular job profile. This system will help the HR department to easily shortlist the candidate based on the CV ranking policy.

This system employs a machine learning algorithm namely "Logistic Regression" which helps to choose fair decisions to recruit a suitable candidate. It has the potential to transform research and assessment in personality psychology. Algorithms can handle vast datasets, including thousands of attributes, without succumbing to collinearity issues. Moreover, ML algorithms are highly efficient in recognizing patterns in datasets that humans cannot even perceive. It does not consume more time to identify the personalities. This predictive personality makes the recruiters understand if a candidate will be a top performer and fit to the culture of the company easily.

V. SYSTEM ARCHITECTURE



VI. DESIGN ALGORITHM

In this section we state the machine learning algorithm (TF-IDF) for CV analysis. The TF-IDF Algorithm is used to find out the important keywords in a document/CV. Below, we give the working of TF-IDF in detail.

Step 1: Calculate TF (Term Frequency) Term Frequency (TF) - Number of times a keyword appeared in a document is calculated by Term Frequency. $TF('keyword') = \text{number of times 'keyword' appears in document} / \text{Total number of keywords in the document}$. Here, the term 'keyword' signifies any job specific skill which the algorithm is searching for.

Step 2: Calculate IDF (Inverse Document Frequency) value.

The problem of rare and frequent words is resolved. This helps our system to give more priority to the required word or skills. IDF sets the log value=1 for the required CV as per skill sets and log value=0 for the unwanted CV.

$IDF('keyword') = \log(\text{total amount of CV} / \text{Number of documents with term keyword})$

Step 3: Calculate TF-IDF weight

$\text{Weight} = TF('keyword') * IDF('keyword')$

Higher the weight, more relevant is the CV and lower the weight, less or no relevance of the CV for the selection process. This step returns the CV with highest and lowest weight values which is further useful for classification.

VII. RESULTS

Admin login:

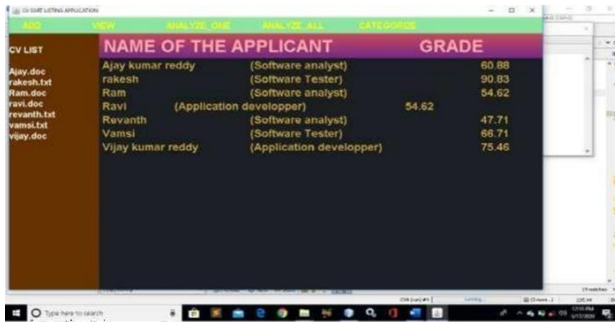


To analyze the cv at first we need to load the file containing the information about the person who want to analyze, we load the any number of cv by just clicking the button add and it shows the popup window and we select the file using the path of the file by file chooser, 1st main idea coming from this section in this section the company provides a basic email and they offer to all persons who are eligible for the job they need to send their resume to the particular email after completion of the time period the company head collect the all resumes and store those files anywhere in the system.

Filechooser:



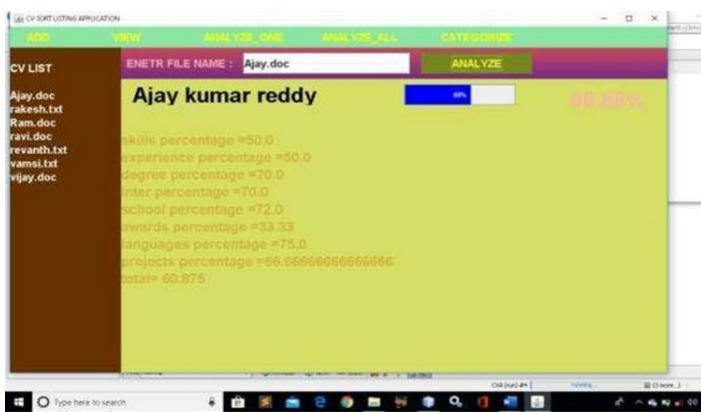
Read The Content:



CV LIST	NAME OF THE APPLICANT	GRADE
Ajay.doc	Ajay kumar reddy (Software analyst)	60.88
rakesh.txt	rakesh (Software Tester)	90.83
Ram.doc	Ram (Software analyst)	54.62
ravi.doc	Ravi (Application developer)	54.62
revanth.txt	Revanth (Software analyst)	47.71
vamsi.txt	Vamsi (Software Tester)	66.71
vijay.doc	Vijay kumar reddy (Application developer)	75.46

This module enables the user or officer to view the content in the document and observe the format of the cv is suitable for your application and this module offers to head officials to read a particular cv and check for any tricks that apply for the grade.

Analyze one cv:



CV LIST	ANALYZE ONE
Ajay.doc	<p>ENTER FILE NAME : Ajay.doc</p> <p>Ajay kumar reddy</p> <p>skills percentage =50.0 experience percentage =50.0 degree percentage =70.0 inter percentage =70.0 school percentage =72.0 awards percentage =33.33 languages percentage =75.0 projects percentage =66.66666666666666 total= 60.875</p>

We can also analyze the one cv at a time by using the module analyze one panel, we simply type the file name in the text field provided in the same panel. After loading the file into the panel it analyze the cv and give the information about the candidate and his overall grade and overview in all fields.

Analyze all:

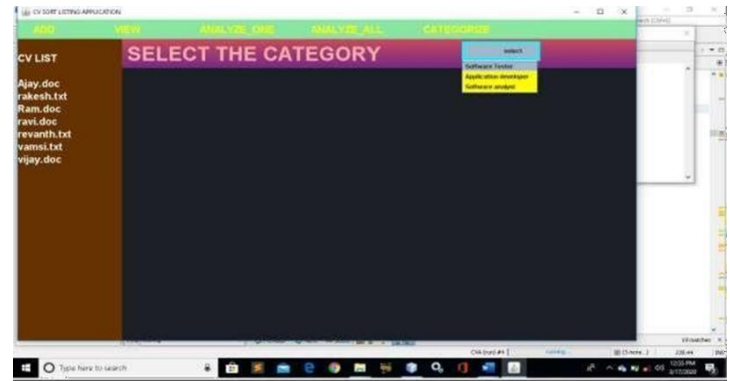


CV LIST	NAME OF THE APPLICANT	GRADE
Ajay.doc	Ajay kumar reddy (Software analyst)	60.88
rakesh.txt	rakesh (Software Tester)	90.83
Ram.doc	Ram (Software analyst)	54.62
ravi.doc	Ravi (Application developer)	54.62
revanth.txt	Revanth (Software analyst)	47.71
vamsi.txt	Vamsi (Software Tester)	66.71
vijay.doc	Vijay kumar reddy (Application developer)	75.46

☐ We can also analyze the all cv we loaded into the application in this window showing all the grades for each applicant.

Categorize:

☐ We can also find out the presongs based on their work category



VIII. CONCLUSION AND FUTURE SCOPE

☐ In recent years, all industries and IT companies feel high burden to analyze all resumes provided by the candidates who are willing to participate job requirement processes so we need automated system for analyzing and shortlisting the all resumes at a time and also calculate the grades of the persons and their individual grades in each section like experience ,skills, education background , languages known and number of projects did. So this project will help almost all companies who are willing to analyze all resumes at a time without putting any human effect. In future cv analysis is required for each and every organization and adding Additional features like “ formats of the resumes are independent for analysis and applying for a job “. This will help to reduce the time taken to analyses all the cv that are received from many candidates who are willing to do the particular job that a company offers . which can predict the traits of an individual using the group of tweets posted by him. It is different from the approaches reviewed in the literature survey, in that it works with the group of tweets and does not take user's profile into account. Further, the system.

IX. REFERENCES

1. Anderson, N. H., & Barrios, A.A. (1961). Primacy effects in personality impression formation. *Journal of Abnormal and Social Psychology*, 63, 346- 350. <https://doi.org/10.1037/h0046719>
2. Arnulf, J., K., Tegner, L., & Larssen, Ø. (2010). Impression making by resume layout: Its impact on the probability of being shortlisted. *European Journal of Work and Organizational Psychology*, 19, 221-230. <https://doi.org/10.1080/13594320902903613>
3. Augustin, H. M. (1991). The written job search: A comparison of the traditional and a nontraditional approach. *Business Communication Quarterly*, 54, 13-14. <https://doi.org/10.1177/108056999105400304>
4. Bendick, M., Jackson, C. W., & Romero, J. H. (1996). Employment discrimination against older workers: An experimental study of hiring practices. *Journal of Aging & Social Policy*, 8, 25-46. https://doi.org/10.1300/J031v08n04_03
5. Bertrand, M., & Mullainathan, S. (2004). Are Emily and Greg more employable than Lakisha and Jamal? A field experiment on labor market discrimination. *The American Economic Review*, 94, 991-1013. <https://doi.org/10.1257/0002828042002561>
6. Bird, C. P., & Puglisi, D. D. (1986). Method of resume reproduction and evaluations of employment suitability. *The Journal of Business Communication*, 23, 31-39. <https://doi.org/10.1177/002194368602300303>
7. Blackburn-Brockman, E., & Belanger, K. (2001). One page or two?: A national study of CPA recruiters' preferences for resume length. *The Journal of Business Communication*, 38, 29-57. <https://doi.org/10.1177/002194360103800104>

DOI: 10.55041/IJSREM32092



ISSN: 2582-3930

Impact Factor: 8.448

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT

An Open Access Scholarly Journal || Index in major Databases & Metadata

CERTIFICATE OF PUBLICATION

International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to

K. Gayathri Venkata Aasritha

in recognition to the publication of paper titled

CURRICULUM VITAE ANALYSIS USING MACHINE LEARNING ALGORITHM

published in IJSREM Journal on Volume 08 Issue 04 April, 2024

Editor-in-Chief
IJSREM Journal

www.ijsrem.com

e-mail: editor@ijsrem.com