SHRI G. S. INSTITUTE OF TECHNOLOGY AND SCIENCE, INDORE

Major Project
Resume Scoring And Filtering

Author:
Khushi Shivde
Mehul Agrawal
Monish Ostwal

Supervisor:
Mrs. Megha Kuliha

A PROJECT REPORT (Phase-I)

submitted to ***Rajiv Gandhi Proudyogiki Vishwavidyalaya, Bhopal***

In partial fulfillment of the requirements for the degree of Bachelor of Engineering

in the

INFORMATION TECHNOLOGY

March 1, 2021

Acknowledgement

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.We are highly indebted to Mrs. Megha Kuliha for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our gratitude towards my parents & members of Shri Govindram seksaria Institute of Science and Technology for their kind cooperation and encouragement which helps us in completion of this project.

We also thanks to our colleagues in developing the project and people who have willingly helped me out with their abilities.

Thanking you,
Khushi Shivde
Mehul Agrawal
Monish Ostwal

Contents

CHAPTER 1

Introduction

1.1 Statement of Project

1.1.1 Need of Resume Scoring and Filtering:

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

1.1.2 Problems and Solution:

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

1.2Motivation

The current recruitment process is more tedious and time consuming which forces the candidates to fill all their skill and information manually. An HR team requires more man power to scrutinize the resumes of the candidates. So that motivated me to build a solution that is more flexible and automated.

1.3 Objective and scope

1.3.1 Objective:

The major objective of our system is to take the current resume ranking system to another level and make it more flexible for both the entities.

1) Candidates, who have been hired.
2) Client company, who is hiring the candidates.

Candidates, who has been hired :

Candidates who are searching for jobs after they graduate. Out of those, a large number of candidates are so desperate that they are ready to work on any post irrelevant to their skill set and ability. The main reason behind this unemployment is like a cancer to our society, if a guy has not got a place after being passed out for 1yr, society includes relatives starting blaming that guy. In Spite of this reason the candidates are ready to work in any condition, on any post. So they don't have to face those situation.

Where our system helps such candidates to get hired by such a company or an organisation who are really worth their ability and their skill sets. Where our algorithm will work in such a way that with the help of the previous result and previous ranking constraints, it will try to optimize the current result, which we called it Machine Learning.

This will make sure that the relevant candidate is hired for that particular vacancy.
You can say the best possible candidate.

Client company, who is hiring the candidates :

Like I am the owner of a particular organisation, obviously my aim would be to create such a team which is the best team in the world. It is like, if there is a vacancy of a java developer in my organisation. So, I won't prefer to hire a python developer and then make him learn Java. That will be pretty useless and time consuming for both that candidate and for the organisation too.

Where our system helps the organisation to make out the best possible candidates list according to their given constraints and requirement for that particular vacancy.

This kind of approach will help our hiring sector to improve like anything and make it more efficient as the relevant person is getting a relevant job. So there would be no regrets for both the entities, client company and that hired candidate. Hence satisfaction will be achieved.

1.3.2 Scope:

As we know Indian I.T sector is the second largest candidate recruiting sector of our country. It contributes about 7.5% to our Gross Domestic Product(G.D.P) Our Proposed system is initially concerned with the I.T sector of our country. It is mainly going to deal with the Indian I.T industry but if you talk about the pro version of our system it can be extended to various other commercial sectors where, intake and elimination are in bulk like for Governmental Jobs.

1.4 System Architecture

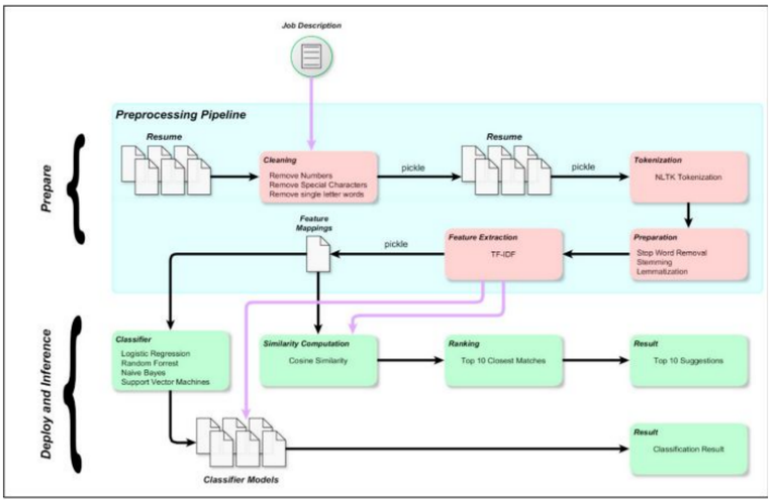


fig1.4.1. System Architecture

CHAPTER 2

Literature Review

2.1 Case Study on talent acquisition

2.1.1 First Generation Hiring Systems

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

2.1.2 Second Generation Hiring Systems

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

2.1.3 Third Generation Hiring Systems

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

Attributes are:

1)Current Compensation
2)Expected Compensation
3)Education
4) Specialization
5) Location
6)Earliest Start Date
7) Work G
8)Total Experience
9)Relevant Experience
10)Communication
11)Current Employer
12)Stability
13)Education Gap

2.2 Weaknesses

1. Prior systems needed a lot of human efforts and time.
2. Cost of hiring is high.
3. Potential candidates may lose the opportunity because of ambiguous keyword matching.
4. Resumes needed to be in specific format.

2.3 How to overcome

1. Use of NLP to read resumes allows candidates the freedom to choose any format that's available to them.

1. Machine learning is used to rank candidates in accordance with requirements Which reduces the efforts of sorting thousands of resumes.

2. Use of NLP can be used to get meaning out of ambiguous data.

3. Five benefits of A.I.

   Goes Beyond Keywords Fast
   and Accurate
   Perfect For the New World of Social Recruiting
   Customizes to your Needs
   Gets Smarter

CHAPTER 3

Requirement Analysis

3.1 Software Requirements

The software requirements in this project include:

- Python
- nltk
- ML

Python:

Python is used for creating backbone structure. Python is intended to be a highly readable language. It is designed to have an uncluttered visual layout, it uses whitespace indentation, rather than curly braces or keywords. Python has a large standard library, commonly cited as one of Python's greatest strengths.

Natural Language Processing Tool :Natural Language Toolkit (NLTK) (Python Package)

NLTK was originally created in 2001 as part of a computational linguistics course in the Department of Computer and Information Science at the University of Pennsylvania. Since then it has been developed and expanded with the help of dozens of contributors. It has now been adopted in courses in dozens of universities, and serves as the basis of many research projects. NLTK was designed with four primary goals in mind:

Simplicity

To provide an intuitive framework along with substantial building blocks, giving users a practical knowledge of NLP without getting bogged down in the tedious house-keeping usually associated with processing annotated language data .

Consistency

To provide a uniform framework with consistent interfaces and data structures, and easily guessable method names .

Extensibility

To provide a structure into which new software modules can be easily accommodated, including alternative implementations and competing approaches to the same task.

Modularity

To provide components that can be used independently without needing to understand the rest of the toolkit.A significant fraction of any NLP syllabus deals with algorithms and data structures. On their own these can be rather dry, but NLTK brings them to life with the help of interactive graphical user interfaces that make it possible to view algorithms step-by-step. Most NLTK components include a demonstration that performs an interesting task without requiring any special input from the user. An effective way to deliver the materials is through interactive presentation of the examples in this book, entering them in a Python session, observing what they do, and modifying them to explore some empirical or theoretical issue.

Machine Learning tool :Scikit-learn (Python Package)

It is a Python module integrating classic machine learning algorithms in the tightly-knit scientific Python world (numpy, scipy, matplotlib). It aims to provide simple and efficient solutions to learning problems, accessible to everybody and reusable in various contexts: machine-learning as a versatile tool for science and engineering.

In general, a learning problem considers a set of n samples of data and try to predict properties of unknown data. If each sample is more than a single number, and for instance a multi-dimensional entry (aka multivariate data), is it said to have several attributes, or features.

We can separate learning problems in a few large categories:

- *Supervised learning* , in which the data comes with additional attributes that we want to predict .This problem can be either:
  –classification: samples belong to two or more classes and we want to learn from already labeled data how to predict the class of unlabeled data. An example of classification problem would be the digit recognition example, in which the aim is to assign each input vector to one of a finite number of discrete categories.
  – regression: if the desired output consists of one or more continuous variables, then the task is called regression. An example of a regression problem would be the prediction of the length of a salmon as a function of its age and weight.
- *Unsupervised learning* , in which the training data consists of a set of input vectors x without any corresponding target values. The goal in such problems may be to discover groups of similar examples within the data, where it is called clustering, or to determine the distribution of data within the input space, known as density estimation, or to project the data from a high-dimensional space down to two or three dimensions for the purpose of visualization

3.2 Hardware Requirements

Linux: GNOME or KDE desktop GNU C Library (glibc) 2.15 or later, 2 GB RAM minimum, 4 GB RAM recommended, 1280 x 800 minimum screen resolution.

Windows: Microsoft R Windows R 8/7/Vista (32 or 64-bit) 2 GB RAM minimum, 4 GB RAM recommended, 1280 x 800 minimum screen resolution, Intel R processor with support for Intel R VT-x, Intel R EM64T (Intel R 64) Execute Disable (XD) Bit functionality.

3.3 Supportive Operating Systems :

The supported Operating Systems for client include:

- Windows xp onwards
- Linux any flavour.

Windows and Linux are two of the operating systems that will support comparative websites. Since Linux is an open source operating system, This system which will be used in this project is developed on the Linux platform but is made compatible with windows too.The comparative website will be tested on both Linux and windows. The supported Operating Systems for server include: The supported Operating Systems For server include Linux. Linux is used as a server operating system. For web server we are using apache 2.0

CHAPTER 4

Project Design

4.1 Design Approach

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical real- ization. Once the software requirements have been analyzed and specified the software design involves three technical activities: design, coding, implementation and testing that are required to build and verify the software. The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer requirements into finished software or a system. Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

4.2 Software Architectural Designs

Our system follows the three tier architecture . First tier consists of GUI, Processing block and the Database.
GUI: The GUI(Graphical User Interface) in our project deals with the interface for the user where the user will login and submit his resume in any **format**(pdf,doc,docx,ect.) and social profiles links. The GUI provides a platform for the user to communicate with the database. It acts as a connector as well **as a communicator** which connects the database and helps in transfer of data between the GUI and the database.
Processing block: Processing block is the block where the actual processing of our project is done. This block connects the gui to the database i.e. it acts as a connector as well as communicator which connects the database and helps in transfer of data between the gui and the database. Its main function is to take input from resumes and social **profiles** of the candidate and parse it to store the information and store it in the structured format(json), and database. After storing this information this system will give output **using a web** application.

Database: Database tier is the tier used for the storage of data. This tier contains all the data that is needed for the processing of the whole project. The data in this tier is related to the student information gathered from his/her resumes and social profiles.

Software Architecture Design :

Component Diagram

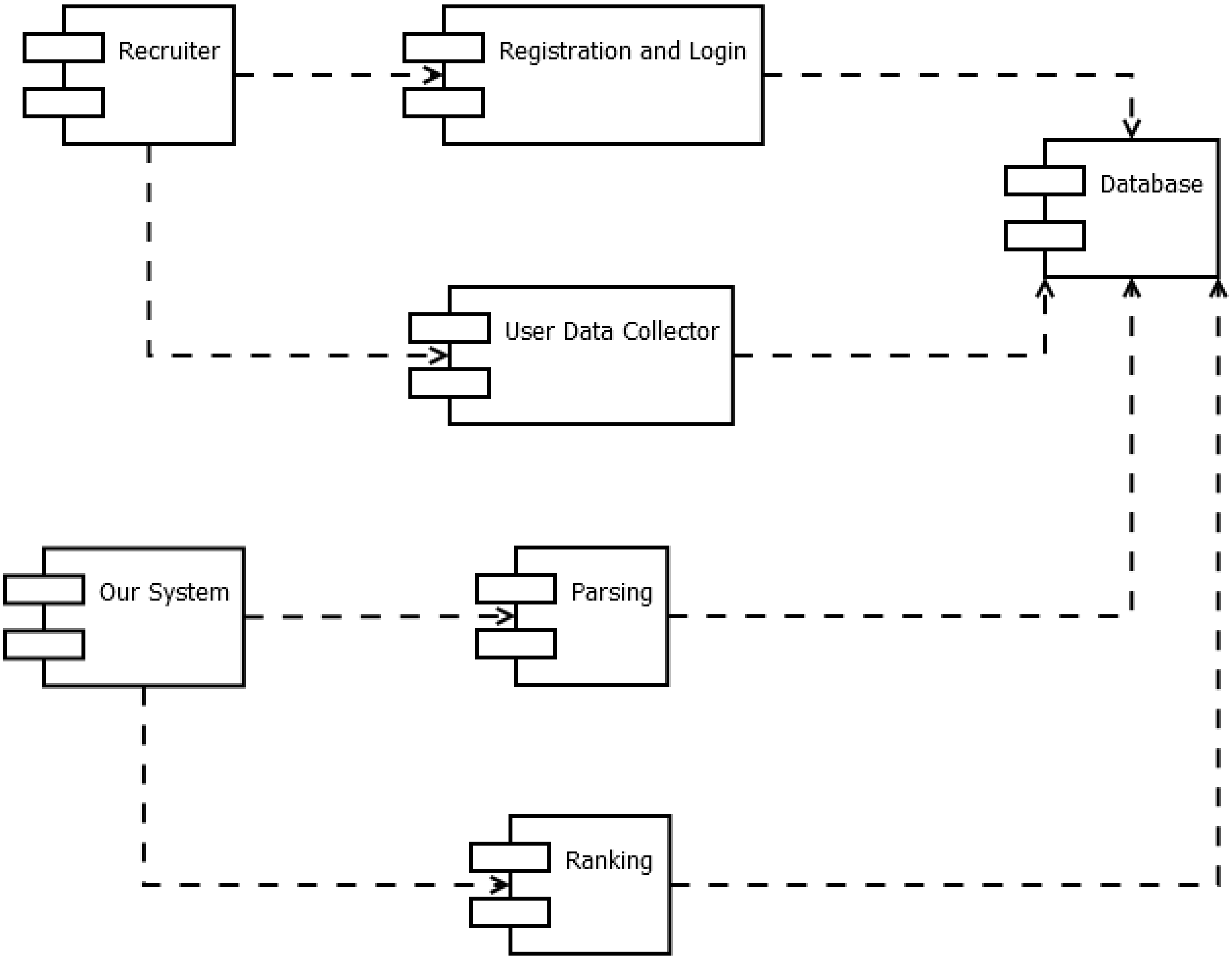


Figure 4.1 - Component Diagram

Use Case

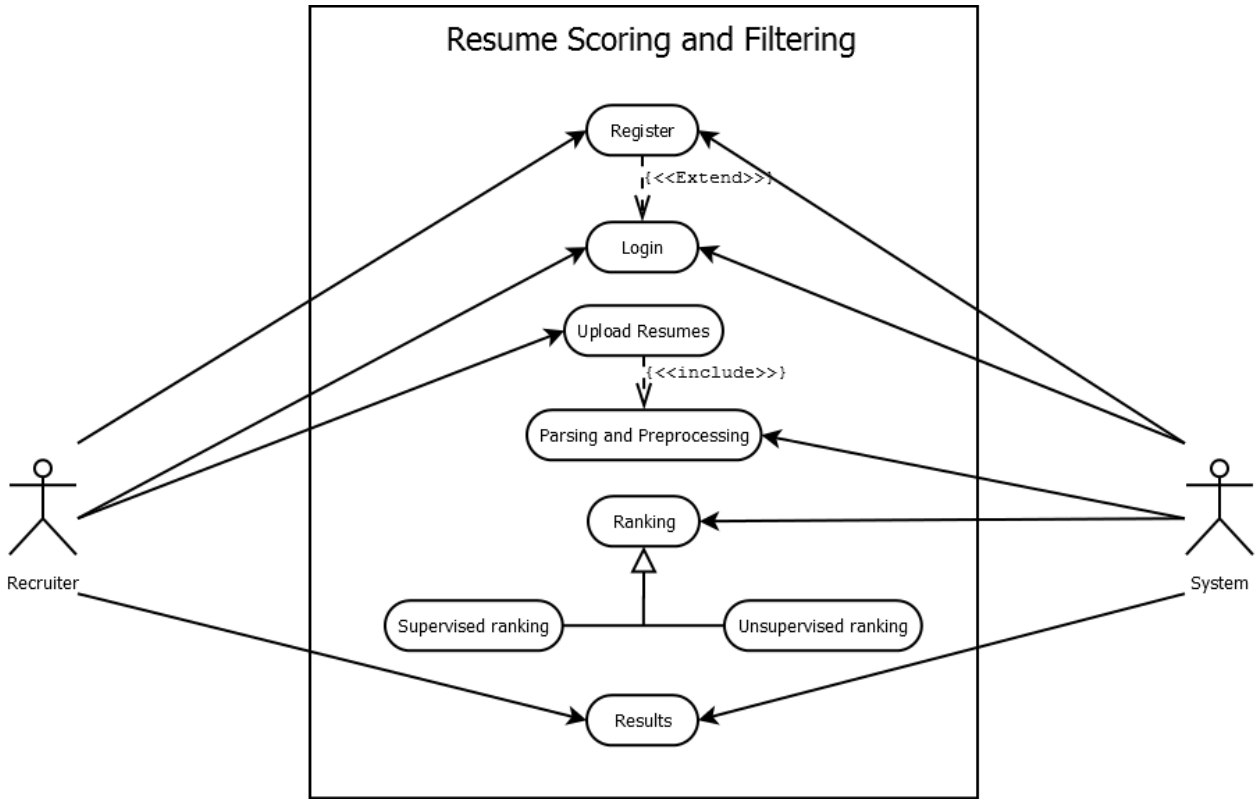


Figure 4.2 - Use Case Diagram

Activity Diagram

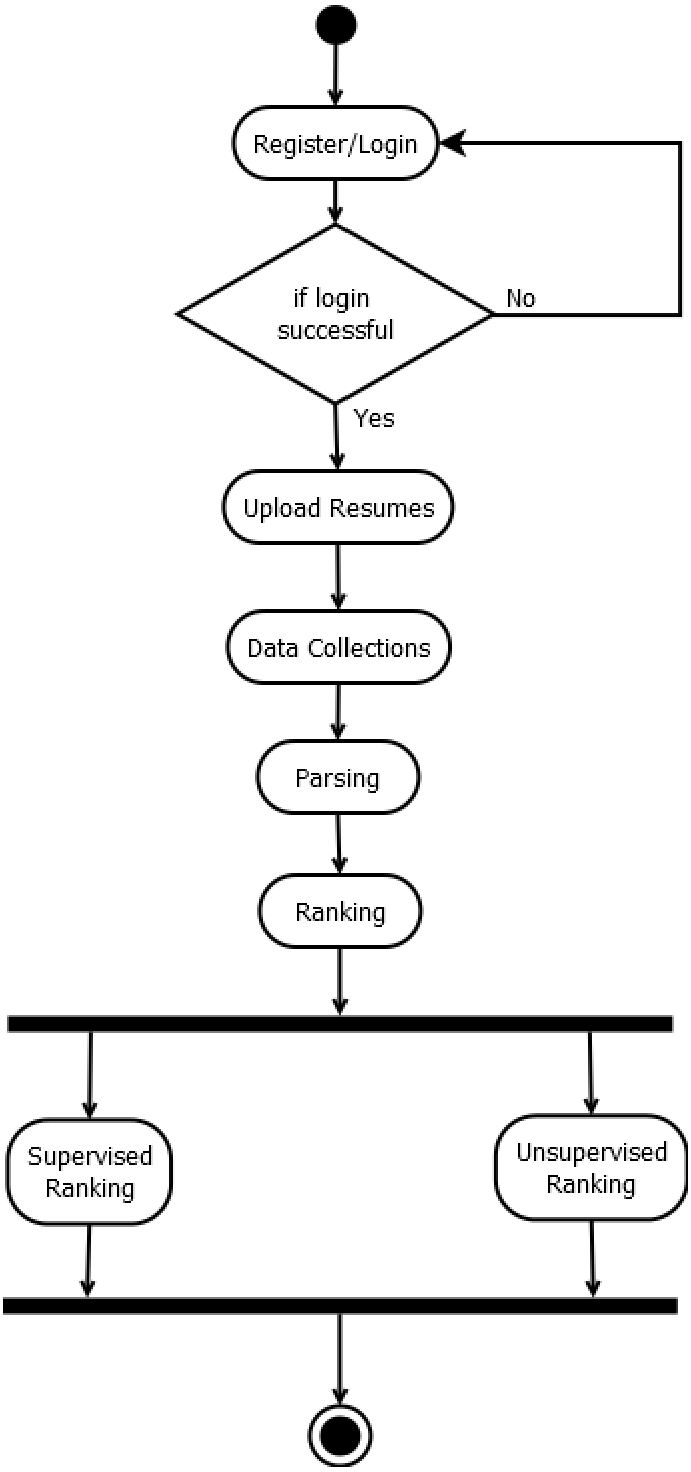


Figure 4.3 - Activity Diagram

Class Diagram

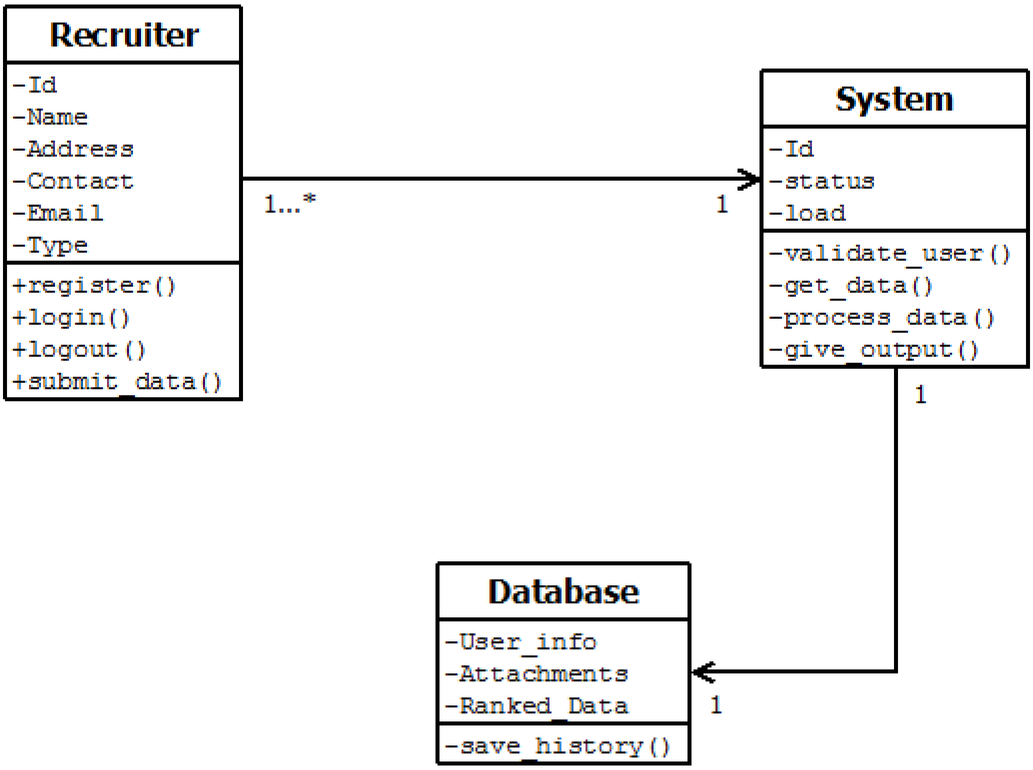


Figure 4.4 - Class Diagram

Sequence Diagram

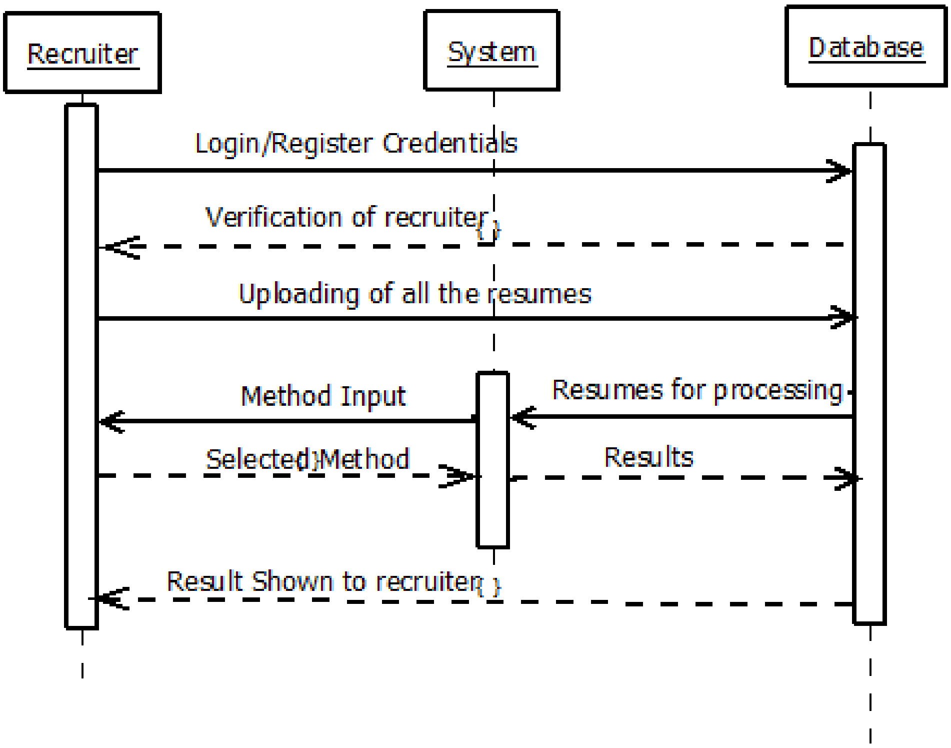

Recruiter
System
Database
Login/Register Credentials
Verification of recruiter
Uploading of all the resumes
Method Input
Resumes for processing
Selected Method
Results
Result Shown to recruiter

Figure 4.5 - Sequence Diagram

ER Diagram

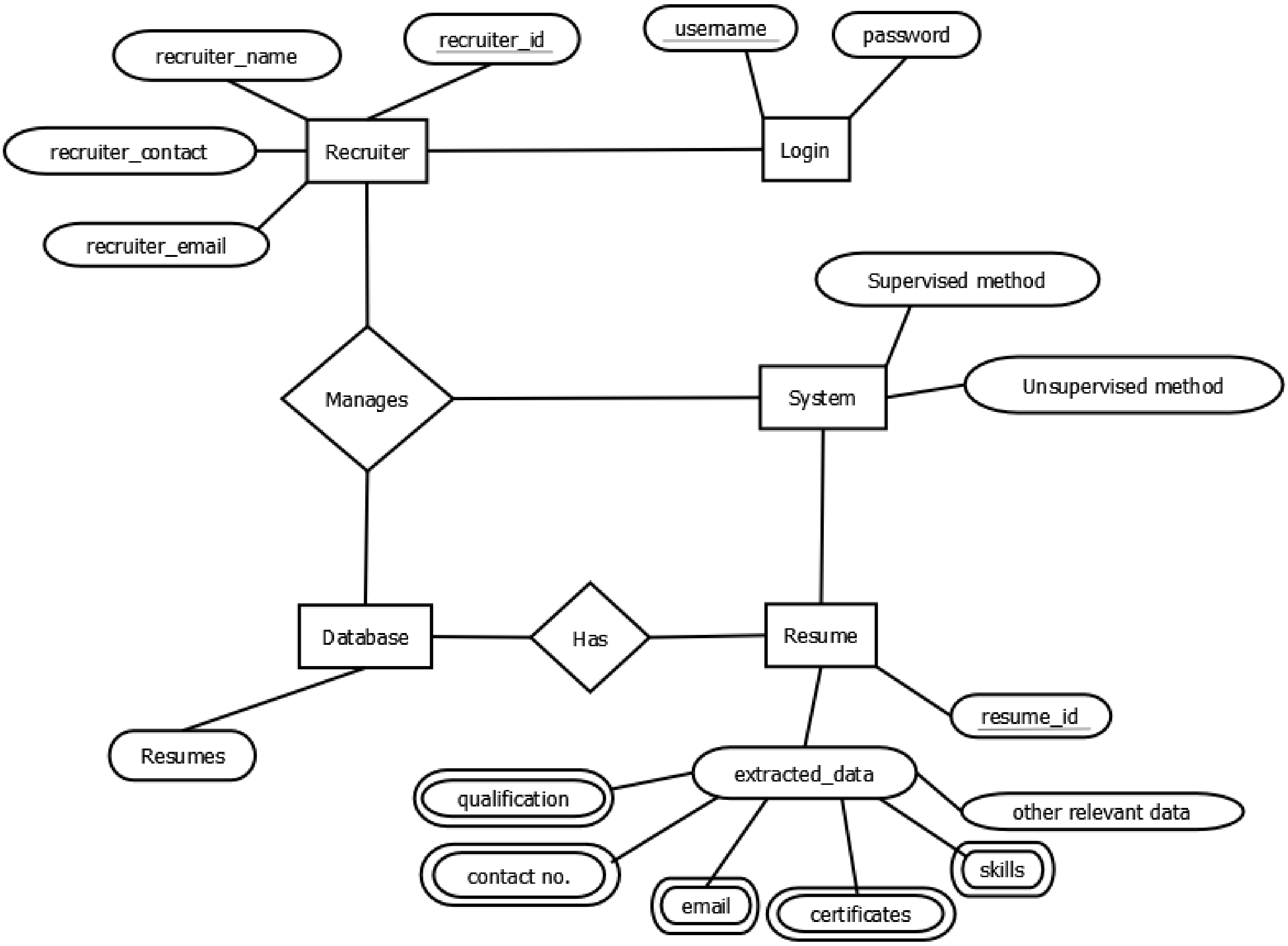
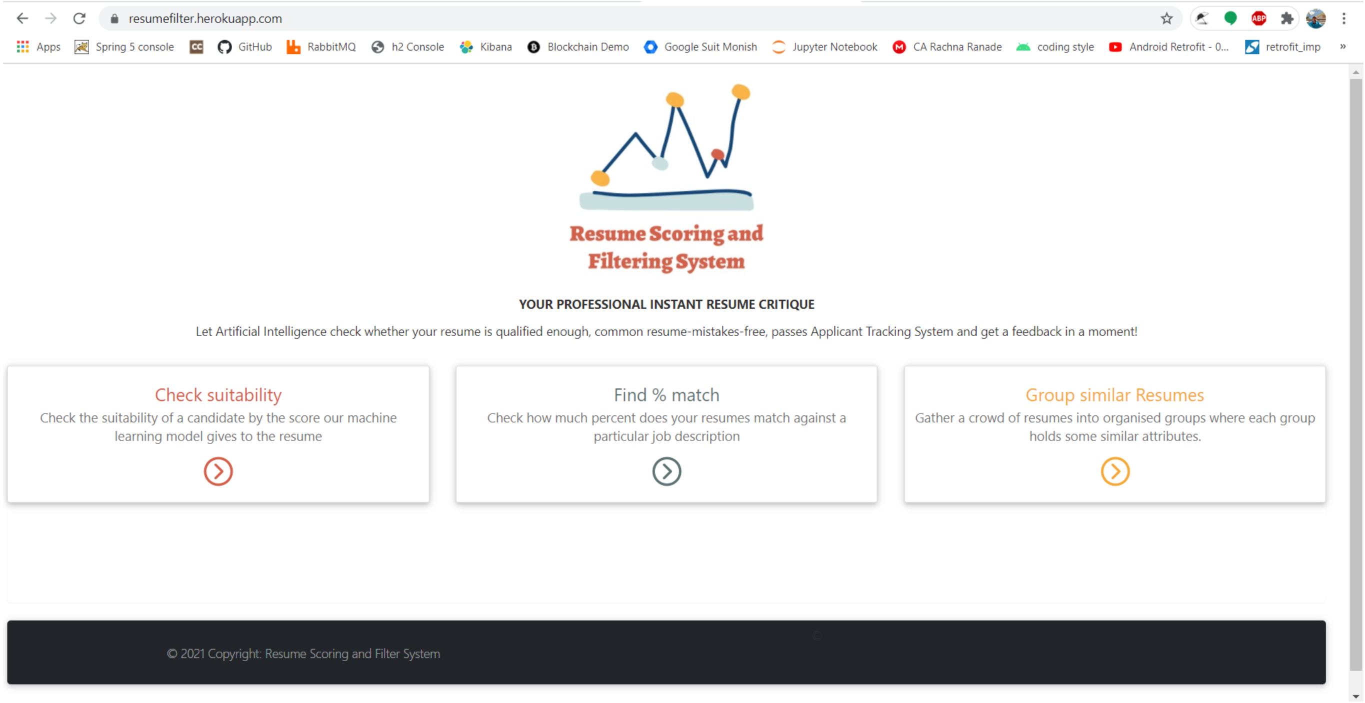


Figure 4.6 - Entity Relationship Diagram

CHAPTER 5

Here are screenshots of the web application interface :

- This will be the homepage of the website.



Resume Scoring and Filtering System

YOUR PROFESSIONAL INSTANT RESUME CRITIQUE

Let Artificial Intelligence check whether your resume is qualified enough, common resume-mistakes-free, passes Applicant Tracking System and get a feedback in a moment!

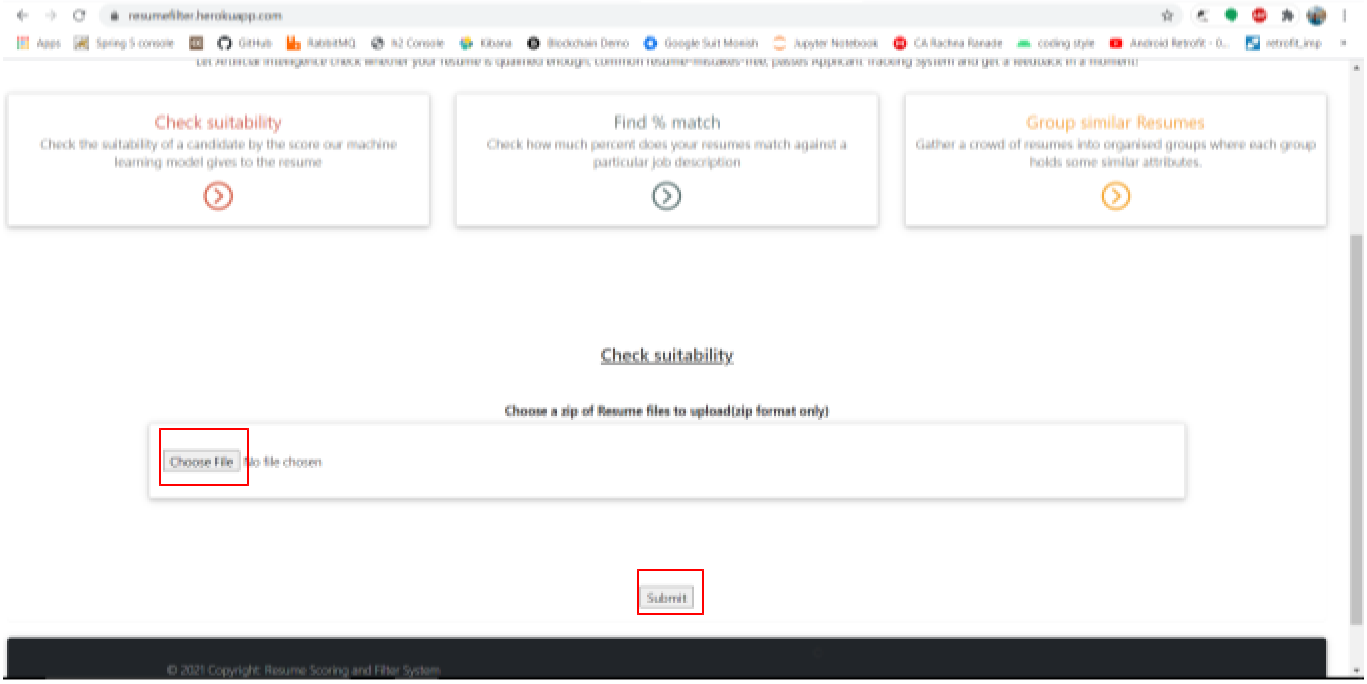
Check suitability
Check the suitability of a candidate by the score our machine learning model gives to the resume

Find % match
Check how much percent does your resumes match against a particular job description

Group similar Resumes
Gather a crowd of resumes into organised groups where each group holds some similar attributes.

© 2021 Copyright: Resume Scoring and Filter System

After selecting any one method you have input corresponding file and click on submit button.



Check suitability
Check the suitability of a candidate by the score our machine learning model gives to the resume

Find % match
Check how much percent does your resumes match against a particular job description

Group similar Resumes
Gather a crowd of resumes into organised groups where each group holds some similar attributes.

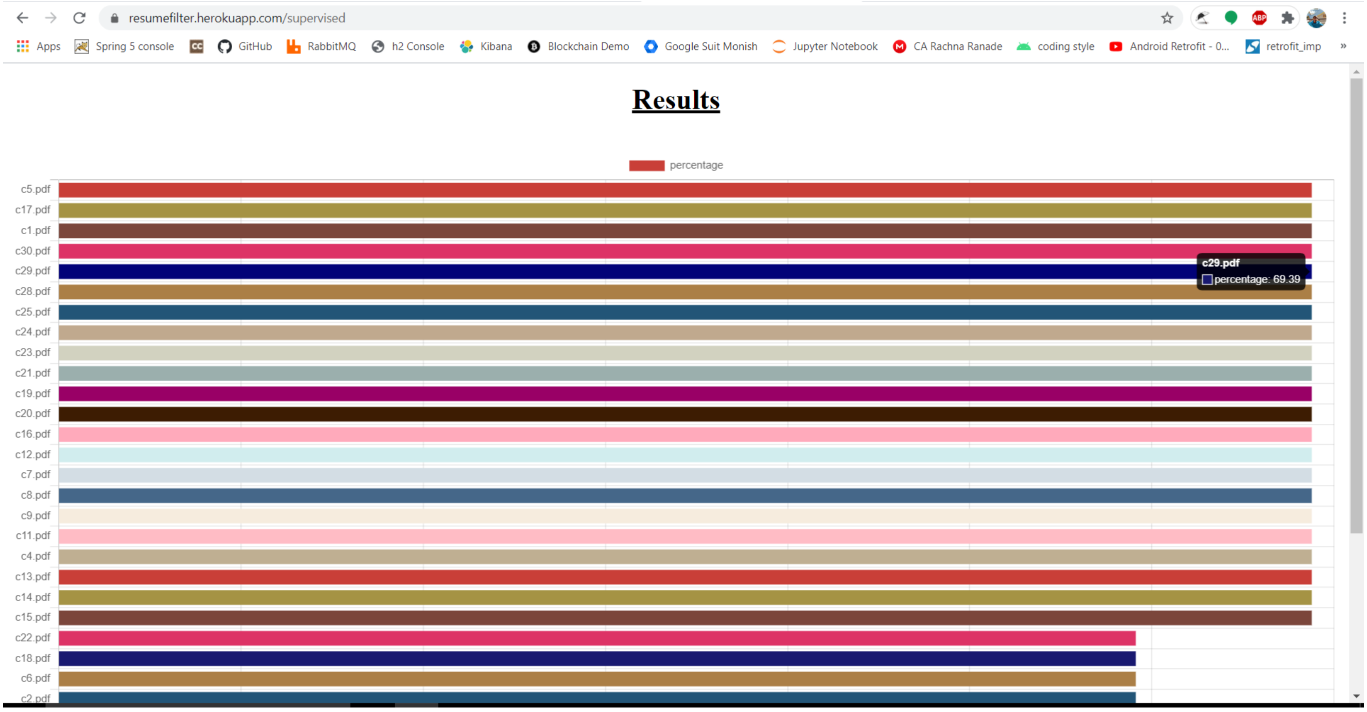
Check suitability

Choose a zip of Resume files to upload(zip format only)

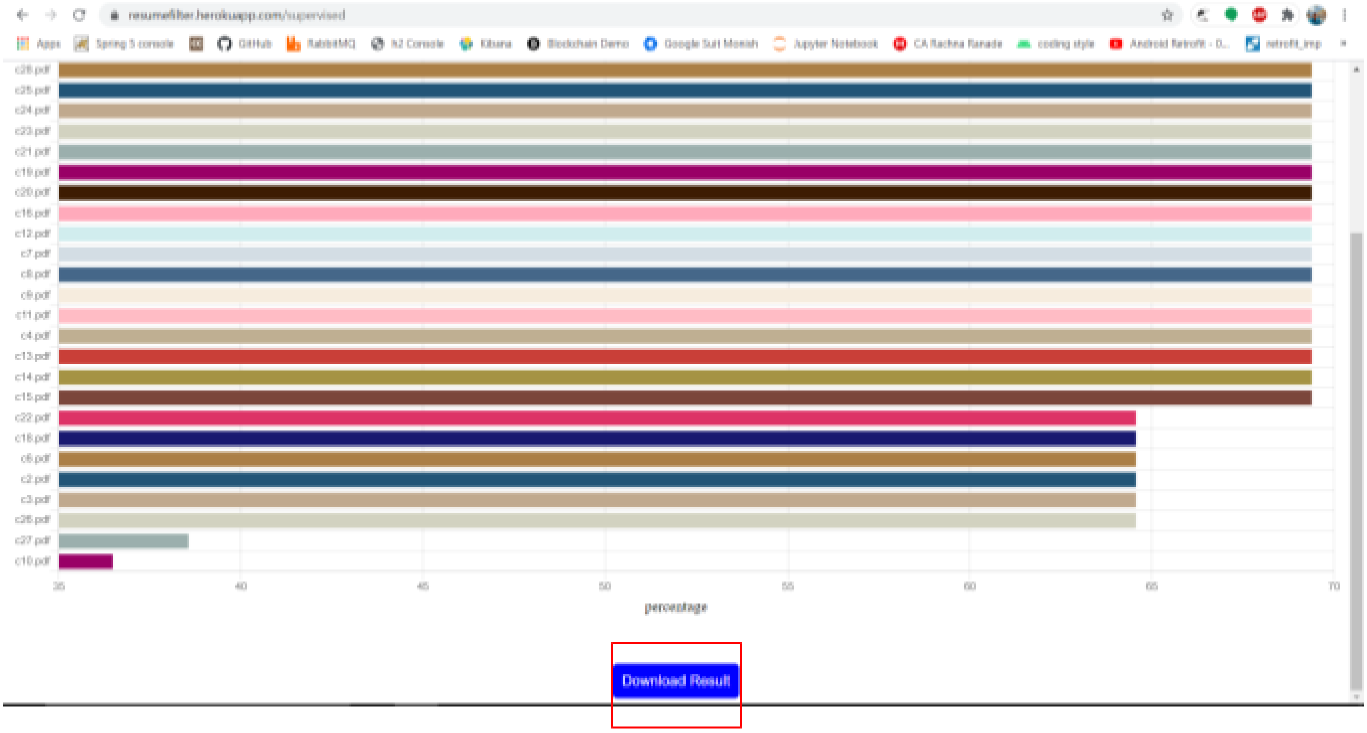
No file chosen

© 2021 Copyright: Resume Scoring and Filter System

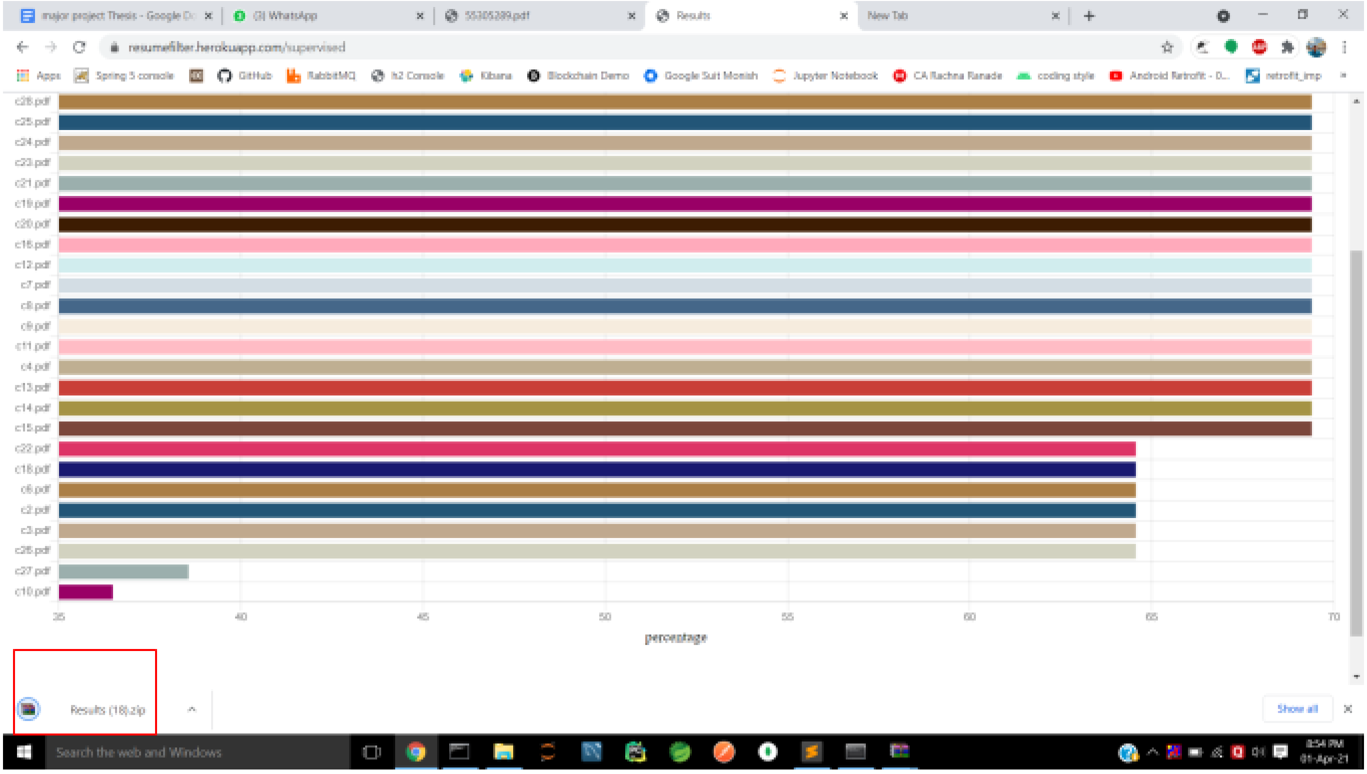
- Graphical result will be shown of the uploaded resume.



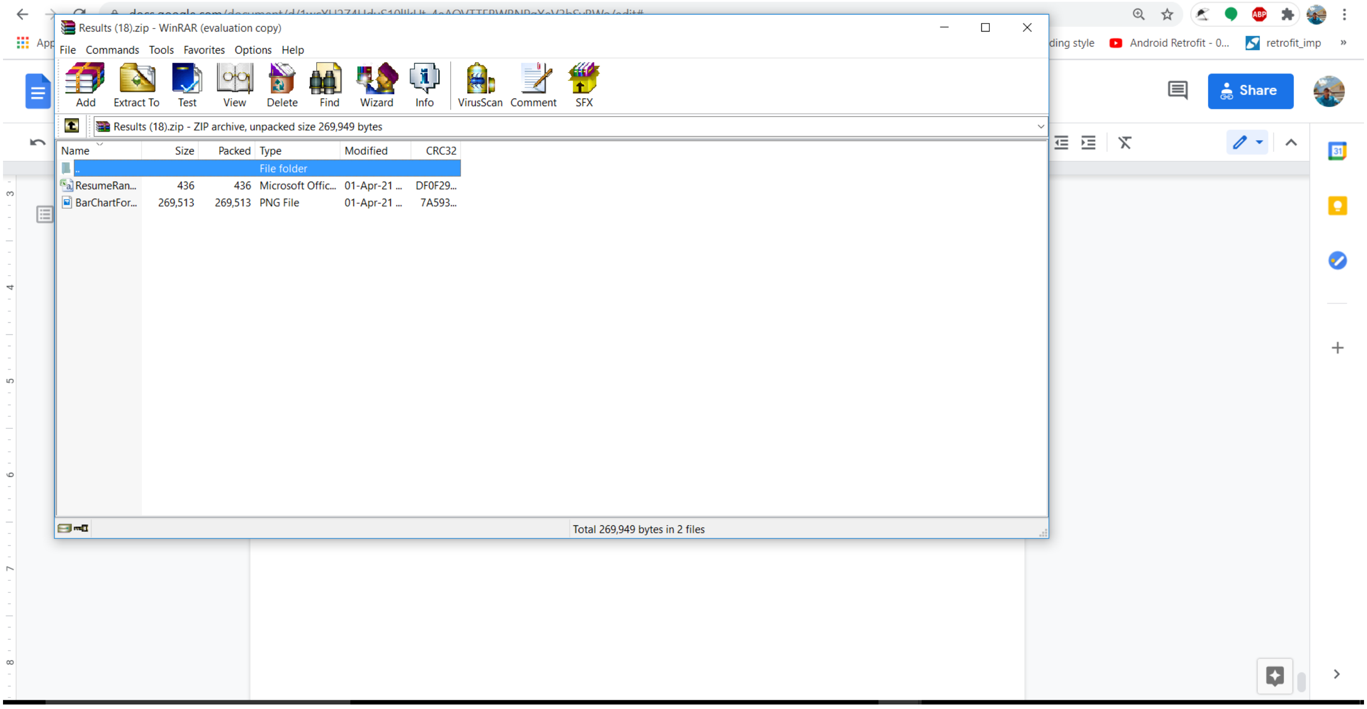
- You can download the result file from click the corresponding button.



- A zip file will be downloaded to your system.



- You can unextract it to see the results of the algorithm chosen by you.



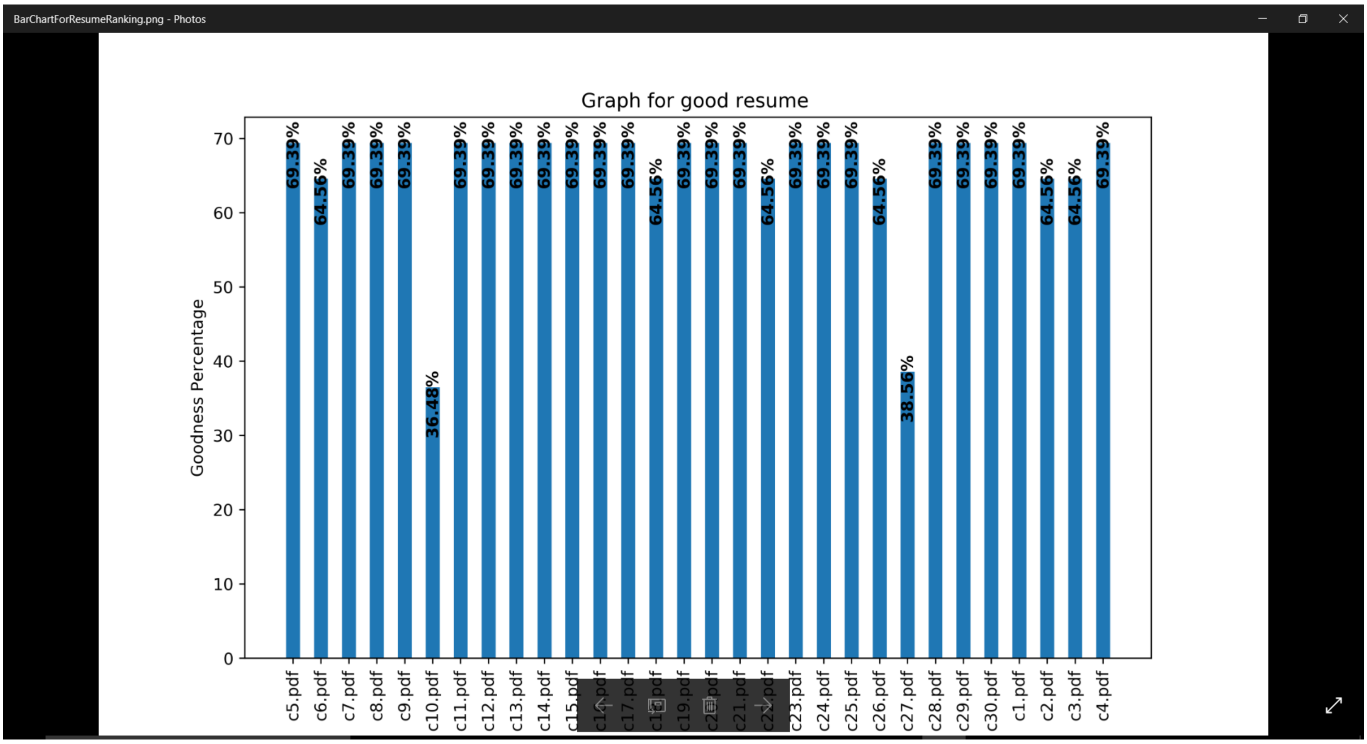
Results (18).zip - WinRAR (evaluation copy)

Results (18).zip - ZIP archive, unpacked size 269,949 bytes

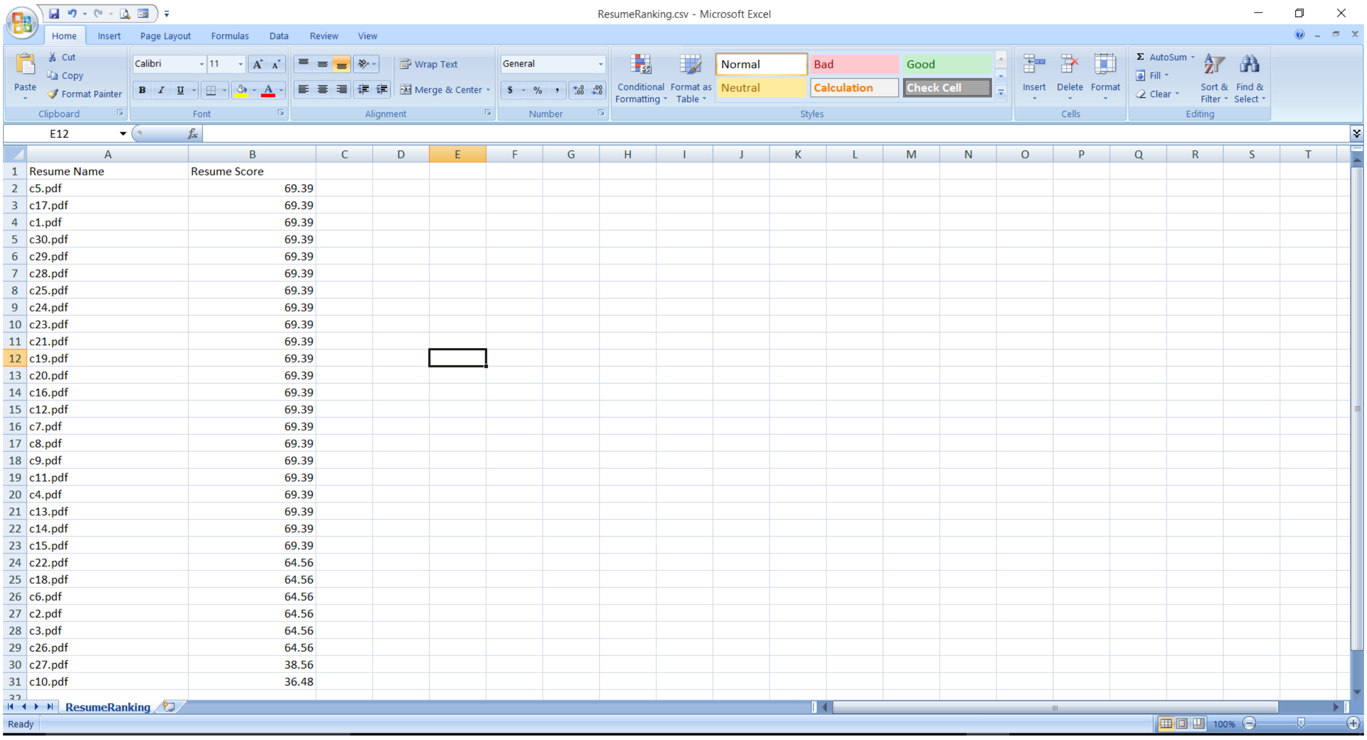
Name	Size	Packed	Type	Modified	CRC32
..			File folder		
ResumeRan...	436	436	Microsoft Offic...	01-Apr-21 ...	DF0F29...
BarChartFor...	269,513	269,513	PNG File	01-Apr-21 ...	7A593...

Total 269,949 bytes in 2 files

- Here is graphical representation of the result.

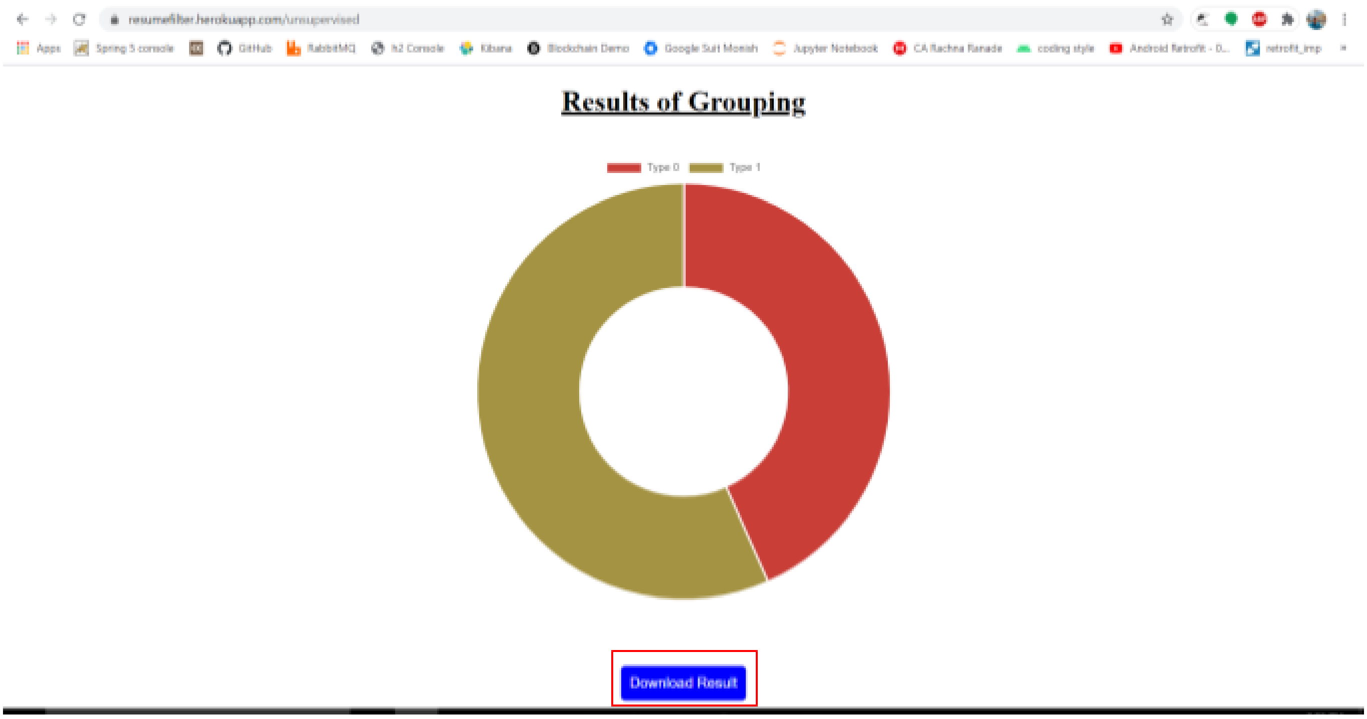


- This will be actual result which would be shown in excel.

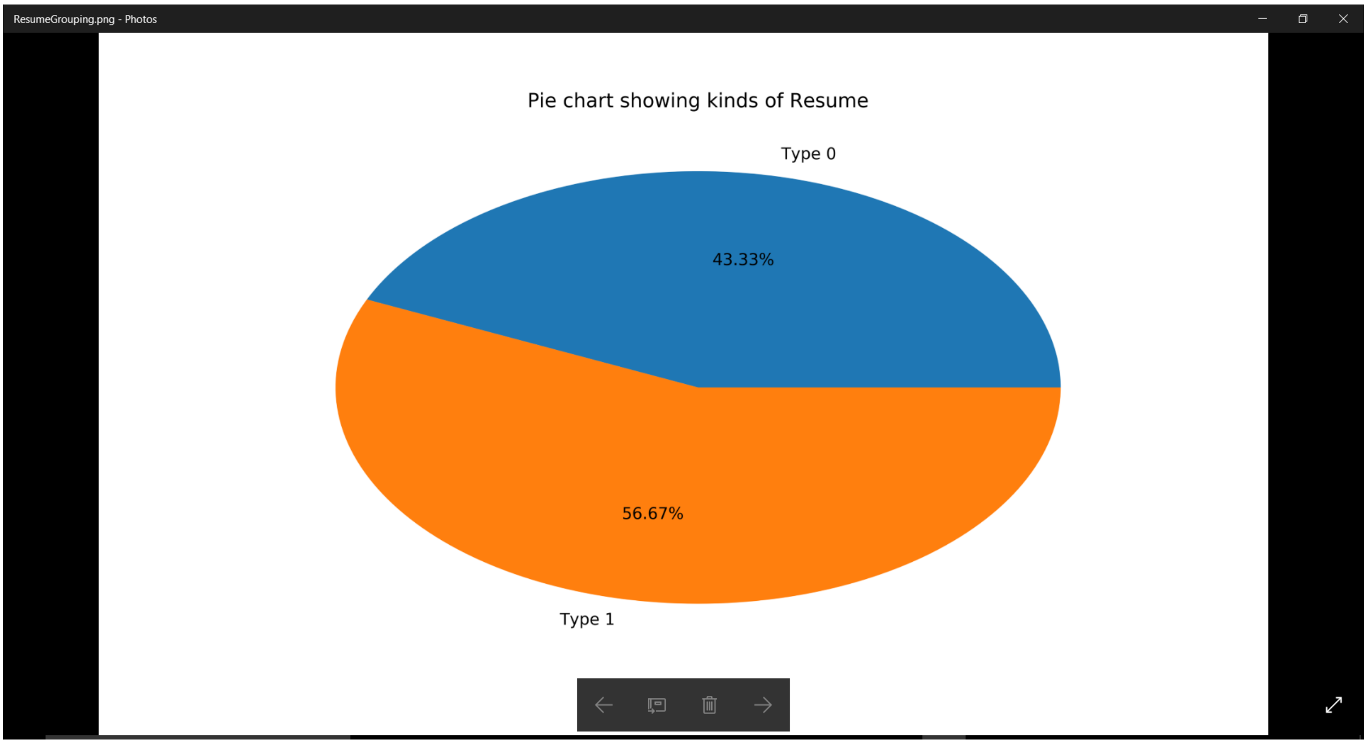


Resume Name	Resume Score
c5.pdf	69.39
c17.pdf	69.39
c1.pdf	69.39
c30.pdf	69.39
c29.pdf	69.39
c28.pdf	69.39
c25.pdf	69.39
c24.pdf	69.39
c23.pdf	69.39
c21.pdf	69.39
c19.pdf	69.39
c20.pdf	69.39
c16.pdf	69.39
c12.pdf	69.39
c7.pdf	69.39
c8.pdf	69.39
c9.pdf	69.39
c11.pdf	69.39
c4.pdf	69.39
c13.pdf	69.39
c14.pdf	69.39
c15.pdf	69.39
c22.pdf	64.56
c18.pdf	64.56
c6.pdf	64.56
c2.pdf	64.56
c3.pdf	64.56
c26.pdf	64.56
c27.pdf	38.56
c10.pdf	36.48

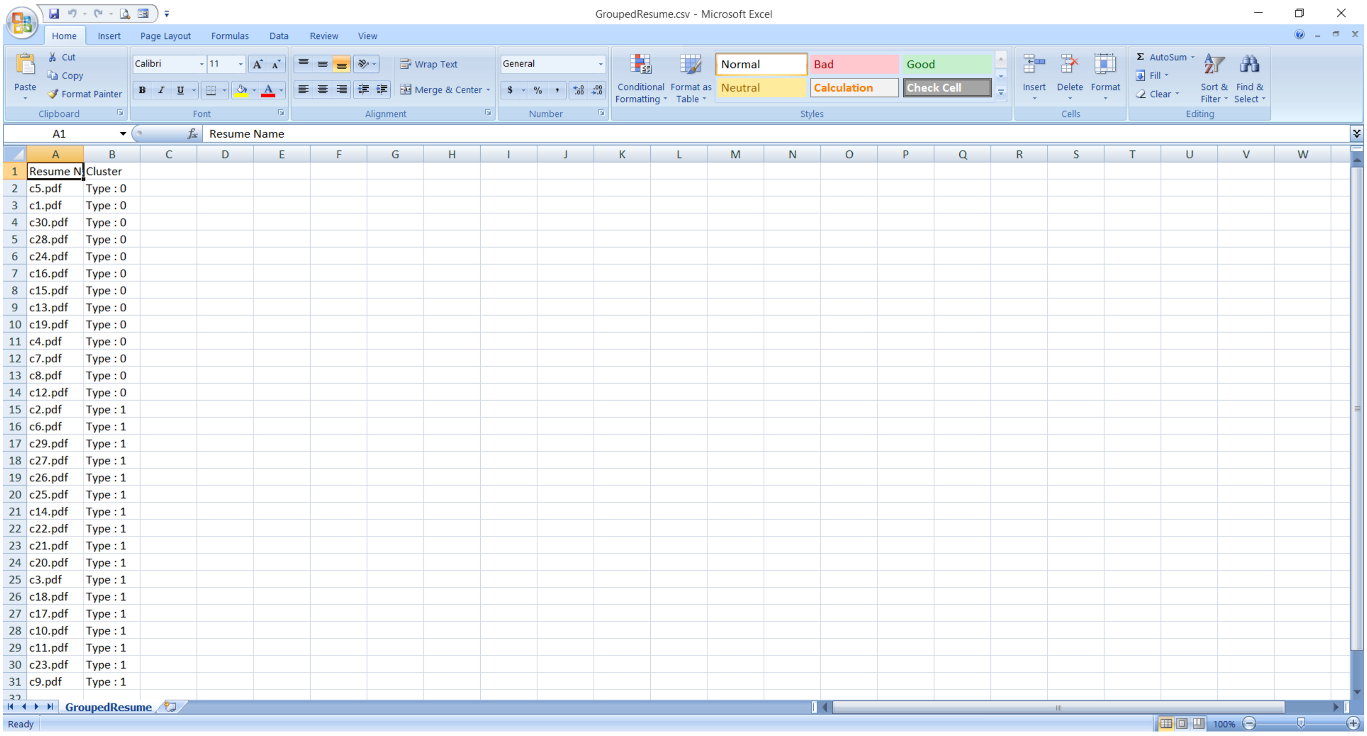
- This is the output for grouping/clustering.



- This graph will be downloaded to your system.


ResumeGrouping.png - Photos
Pie chart showing kinds of Resume
Type 0
43.33%
56.67%
Type 1

- This is corresponding excel file which will show different types of resume.



A1 Resume Name

	A	B
1	Resume N	Cluster
2	c5.pdf	Type : 0
3	c1.pdf	Type : 0
4	c30.pdf	Type : 0
5	c28.pdf	Type : 0
6	c24.pdf	Type : 0
7	c16.pdf	Type : 0
8	c15.pdf	Type : 0
9	c13.pdf	Type : 0
10	c19.pdf	Type : 0
11	c4.pdf	Type : 0
12	c7.pdf	Type : 0
13	c8.pdf	Type : 0
14	c12.pdf	Type : 0
15	c2.pdf	Type : 1
16	c6.pdf	Type : 1
17	c29.pdf	Type : 1
18	c27.pdf	Type : 1
19	c26.pdf	Type : 1
20	c25.pdf	Type : 1
21	c14.pdf	Type : 1
22	c22.pdf	Type : 1
23	c21.pdf	Type : 1
24	c20.pdf	Type : 1
25	c3.pdf	Type : 1
26	c18.pdf	Type : 1
27	c17.pdf	Type : 1
28	c10.pdf	Type : 1
29	c11.pdf	Type : 1
30	c23.pdf	Type : 1
31	c9.pdf	Type : 1