A Project Report on

CV FILTERING TOOL

Submitted to

Jawaharlal Nehru Technological University, Hyderabad

in partial fulfillment of requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

Harini Popuri(16BD1A05A8)
S.Kavya Sumanika(16BD1A05BT)
Pooja Ailani(16BD1A05A6)
K.Prudhvi Teja(16BD1A05AB)

Under the esteemed guidance of
GUIDE NAME

(Assistant/Associate) Professor
Department of CSE

CV Filtering

Department of Computer Science and Engineering

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTUH

3-5-1206, Narayanaguda, Hyderabad - 500029

2019 – 2020

KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY

Approved by AICTE, Affiliated to JNTU, Hyderabad

3-5-1206, Narayanaguda, Hyderabad - 500029.

CERTIFICATE

This is to certify that the project entitled **CV FILTERING TOOL** being submitted by **Ms. HARINI POPURI (16BD1A05A8),Ms.KAVYA SUMANIKA (16BD1A05BT), Ms.POOJA AILANI(16BD1A05A6),Mr.K.PRUDHVI TEJA(16BD1A05AB)** students

of **Keshav Memorial Institute of Technology, JNTUH** in partial fulfillment of the requirements of the award of the Degree of **Bachelor of Technology in Computer Science and Engineering** as a specialization is a record of bonafide work carried out by them under my guidance and supervision in the academic year 2019 – 2020

GUIDE	**HEAD OF THE DEPARTMENT**
Mr./Ms./Dr. NAME OF GUIDE	**Dr. S. Padmaja**
(Assistant/ Associate) Professor	HoD-CSE

CV Filtering

Submitted for the Project Viva Voce examination held on …….

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project report entitled **—CV FILTERING TOOL** is done in the partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering affiliated to Jawaharlal Nehru Technological University, Hyderabad. This project has not been submitted anywhere else.

HARINI POPURI (16BD1A05A8)

S.KAVYA SUMANIKA (16BD1A05BT)

POOJA AILANI(16BD1A05A6)

K.PRUDHVI TEJA(16BD1A05AB)

ACKNOWLEDGEMENT

We take this opportunity to thank all the people who have rendered their full support to our project work.

We render our thanks to **Dr. Maheshwar Dutta**, B.E., M Tech., Ph.D., Principal who encouraged us to do the Project.

We are grateful to **Mr. Neil Gogte**, Director for facilitating all the amenities required for carrying out this project.

We express our sincere gratitude to **Mr. S. Nitin**, Director and **Mrs. Deepa Ganu**, Dean Academics for providing an excellent environment in the college.

We are also thankful to **Dr. S. Padmaja**, Head of the Department for providing us with both time and amenities to make this project a success within the given schedule.

We are also thankful to our guide **Dr.Narendar** for his/ her valuable guidance and encouragement given to us throughout the project work.

We would like to thank the entire CSE Department faculty, who helped us directly and indirectly in the completion of the project.

We sincerely thank our friends and family for their constant motivation during the project work.

CV Filtering

HARINI POPURI (16BD1A05A8)

S.KAVYA SUMANIKA (16BD1A05BT)

POOJA AILANI(16BD1A05A6)

K.PRUDHVI TEJA(16BD1A05AB)

TABLE OF CONTENTS

Content	Page no.
Declaration	iii
Acknowledgement	iv
Abstract	1
List of figures	2
1.INTRODUCTION	3
1.1 Overview of the Project	3
1.2 Aim of the Project	3
1.3 Problems with existing system	3
1.4 Proposed System	4
1.5 Contents	4
2. REQUIREMENT ANALYSIS AND SPECIFICATION	6
2.1 Introduction	6
2.2 Functional Requirements	6
2.3 Non Functional Requirements	6

2.4 Hardware and Software Requirements 7

3. LITERATURE SURVEY 8

3.1 About the project 8

3.2 Benifits 11

3.3 Disadvantages 11

4.TECHNOLOGIES USED 13

4.1 Python 13

4.1.1 Modules 13

4.2 The imaplib library 14

4.3 The smtplib library 16

4.4 The mammoth package 22

4.5 The PyPDF2 package 23

5. SYSTEM DESIGN 25

Use case Diagrams 25

6.IMPLEMENTATION 26

6.1 Flow Chart 27

6.2 Functions 28

7.TESTING AND RESULTS 33

7.1 Testing Objectives 33

7.2 Output Screens 34

8. CONCLUSION AND FUTURE SCOPE 38

9.BIBILIOGRAPHY 39

ABSTRACT

In the present day situation, a company needs to shortlist only few members from a large number of applicants. It is very time consuming and tedious process. The number of applicants, applying for a job, also increases day by day.

Our idea is to develop a project for companies which will enable a more effective way to short list submitted candidate CVs by providing a consistent and fair CV ranking policy. The 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 kind of application plays an important role in simplifying the recruitment process. The system has facilities where candidates can upload the CV's and other academic achievements. Recruitment application make possible for managers to access information that is crucial to managing their stuff, which they can see for human resource management, staffing and planning activities. The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be selected by theirs application based on their qualification and company requirement.

Our project will have two modules, one for admin and another for candidate. Candidate here will register him/herself with all its details and will upload their own CV into the system. The Company can also put an online test to enhance the filtering of CVs.

***Key Words*:** Information extraction, Filtering, Ranking, Shortlisting, Sorting.

LIST OF FIGURES

Figure Name	Page no.
Fig.5.1-Flow Chart	18
Fig.6.1-The terminal during execution	25
Fig.6.2-The folder with downloaded resumes	25
Fig.6.3-The excel sheet with candidate details	26
Fig.6.4-Response mail to selected candidate	26
Fig.6.5-Response mail to rejected candidate	26

CHAPTER-1

1.INTRODUCTION

Overview of the Project:

In the present day situation, a company needs to shortlist only few members from a large number of applicants. It is very time consuming and tedious process. The number of applicants, applying for a job, also increases day by day. So there is a dire need for filtering the resumes and selecting the candidates for further rounds.

Aim of the Project:

Our idea of the project is to automate the process of selecting candidates for further rounds by scanning their resumes sent to the company mail id. The user should give the company mail id and password. Once they are scanned, the following steps take place:

- Logs in into the company mail id.
- Searches for all the unread mails whose subject contains the word ‘resume’ .
- Downloads the attachments(i.e. resumes) and stores the sender(candidate) information.
- Scans them and decides whether the candidate is selected or rejected.

- Sends a response(selected or rejected) to the candidate.

Problems with Existing System:

Existing system performs cutthroat process that generally leaves the companies with a good, but not great employee at a significant financial investment. As the industries have grown, their hiring needs has rapidly grown. To serve these hiring needs certain consultancy units like employment websites have come into existence. They offer a solution in which the candidate has to upload their information and submit it to the website. Then these websites would search the candidates based on certain keywords. These websites are middle level organizations between the candidate and recruiter.

This kind of application plays an important role in simplifying the recruitment process. The system has facilities where candidates can upload the CV's and other academic achievements. Recruitment

application make possible for managers to access information that is crucial to managing their stuff, which they can see for human resource management, staffing and planning activities. The primary purpose to develop this system is to optimize the recruitment process for an organization. Besides, the qualified applicants could be selected by theirs application based on their qualification and company requirement.

These websites are also not flexible as the candidate has to upload there resume in a particular layout, and these formats changed from system to system. These systems charge a certain amount per resume. According to the survey Monster.com charges Rs.1.44 lac (INR) for 50,000 resumes. Hence such systems are not cost effective.

Proposed System:

The motivation behind this system is to develop an application that will assist organizations in the recruitment process. This is far different approach than employment websites. Our system allow the

candidates to enter information about academics, skill set etc. and upload their resumes. The entered information is then analysed by our system. This makes our search process easy. The analysing system works on the algorithm that uses ranking, which is a sub domain of Text Mining. System reads the information entered by user such as SSC marks, HSC marks, degree aggregate, programming languages known and performs ranking. This acquired information is stored in the database. This stored information can be accessed by HR, HR can simply provide keywords to the system and system will find all the relevant resumes that match with the keywords

Contents:

- **Chapter 1** deals with the introduction of the project.
- **Chapter 2** deals with the requirement analysis and specification.
- **Chapter 3** deals with the Literature Survey done for the project.
- **Chapter 4** describes the technologies used for the project.
- **Chapter 5** deals with the implementation of the project.
- **Chapter 6** deals with the testing and results of the project.
- **Chapter 7** deals with the conclusion and future scope of the project.

REQUIREMENT ANALYSIS AND SPECIFICATION

Introduction

The requirements specification is a technical specification of requirements for the soft- ware products. It is the first step in the requirements analysis process it lists the requirements of a particular software system including functional, performance and security requirements. The requirements also provide usage scenarios from a user, an operational and an administrative per-

spective. The purpose of software requirements specification is to provide a detailed overview of the software project, its parameters and goals. This describes the project target audience and its user interface, hardware and software requirements. It defines how the client, team and audience see the project and its functionality.

Functional Requirements:

Functional software requirements help to capture the intended behavior of the system. This behavior may be expressed as functions, services or tasks or which system is required to perform. The functional requirements in this project are:

- The software should be able to download the resumes and scan them after logging into the company's email id.
- It should scan the resumes based on a fair ranking policy and decide whether the candidate is selected or not.
- It should send automated responses to the selected and unselected candidates.

Non-Functional Requirements:

They are essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. The non-functional requirements in this project are:

- Periodically the resumes should be checked from time to time and the response should be sent.
- The program should be active only when the company has some job requirements.
- It should be capable enough to handle a number of resumes without affecting its performance
- The software should be portable. So moving from one OS to other OS does not create any problem.
- Security requirements ensure that the software is protected from unauthorized access to the system and its stored data.
- Privacy of information should be ensured and the export of restricted, intellectual property rights, etc. should be audited.

Hardware and Software Requirements:

Operating System	Windows XP or Mac OSX
Programming Language	PYTHON3
Processor	Dual core processor 2.2GHz or higher(64 bit)
RAM	1 GB or more
Disk Space	2GB recommended

LITERATURE SURVEY

About the project:

The rapid development of modern Information and Communication technologies (ICTs) in the past few years and their introduction into people's daily lives has led to new circumstances at all levels of their social environment (work, interpersonal relations, entertainment, etc). People have been steadily turning to the web for job seeking and career development, using web 2.0 services like LinkedIn and job search sites (Bizer, 2005). On the other hand, a lot of companies use online knowledge management systems to hire employees, exploiting the advantages of the World Wide Web. These are termed e-recruitment systems and automate the process of publish- ing positions and receiving CVs. The online recruitment problem is two-sided: It can be seeker- oriented or company-oriented. In the first case, the system recommends to the candidate a list of job positions that better fit his profile. In the second case recruiters publish the specifications of available job positions, and the candidates can apply, submitting their CVs. Many approaches can be applied to automate the e-recruitment process combining techniques from classical IR (Kessler, 2009). These include collaborative filtering techniques (Rafter, 2000), relevance feed- back (Kessler, 2009), semantic matching (Mochol, 2007), multi-agent systems (De Meo, 2007) etc. Their main drawback comes from the fact that the CVs in these works are either submitted by the user in an arbitrary format or are mined automatically from the Web or other sources (i.e. from server logs).

In 2014 an Integrated E-Recruitment System for Automated Personality Mining and Applicant Ranking was proposed by Faliagka et al. an automated candidate ranking was implemented by this system. It was based on objective criteria that the candidate's details would be extracted from the candidate's LinkedIn profile. The candidates' personality traits were automatically extracted from their social presence using linguistic analysis. The candidate's rank was derived from individual selection criteria using Analytical Hierarchy Process (AHP), while their weight was controlled by the recruiter (admin). The limitations of the system were that senior positions that required expertise and certain qualifications were screened inconsistently.

Liden et al. published The General Factor of Personality: The interrelations among the Big Five personality factors (Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism) were analyzed in this paper to test for the existence of a GFP. The meta-analysis provides evidence for a GFP at the highest hierarchal level and that the GFP had a substantive component as it is related to supervisor-rated job performance were concluded by this paper. However, it is also realized that it is important to note that the existence of a GFP did not mean that other personality factors that were lower in the hierarchy lost their relevance.

Most large corporations utilize C.V Filtering tool. Kelly OCG survey estimated 66% of large companies and 35% of small organizations rely on recruitment software.

There are dozens of different types of this tool, each with their own features, strengths weaknesses and quirks. Here are some of them:

Taleo	Workday	SuccessFactors
Brassring	iCIMS	Jobvite
SmartRecruiters	JazzHR	CATS

Fortune 500 C.V Filtering tool market share

Research shows that at least 491 Fortune 500 companies use C.V Filtering tool. In other words, 96.4% of Fortune 500 companies utilize this software that is found via a variety of methods. The easiest way to identify type of C.V Filtering tool is to look at the URL of a job listing.

Here’s a small sampling of the data:

Company	CV Filtering Tool	Sample Job Listing URL
3M	Workday	https://3m.wd1.my**workday**jobs.com/Search/job/…
Allstate	iCIMS	https://careers-agencystaffcandidate.**iCIMS**.com/…
Clorox	ADP	https://recruiting.**ADP**.com/srccar/public/…
Coca-Cola	Workday	https://coke.wd1.my**workday**jobs.com/coca-cola…
Lockheed Martin	BrassRing	https://sjobs.**BrassRing**.com/TGnewUI/Search/…
Microsoft	Proprietary	https://careers.microsoft.com/us/en/job/…
Nike	Taleo	https://nike.**taleo**.net/careersection/10020/…
Oshkosh	SuccessFactors	https://career4.**SuccessFactors**.com/career?…
United Natural Foods	SilkRoad	https://unfi-openhire.**SilkRoad**.com/epostings/…
Wynn Resorts	CareerBuilder	https://application.**careerbuilder**1.com/wynn-r

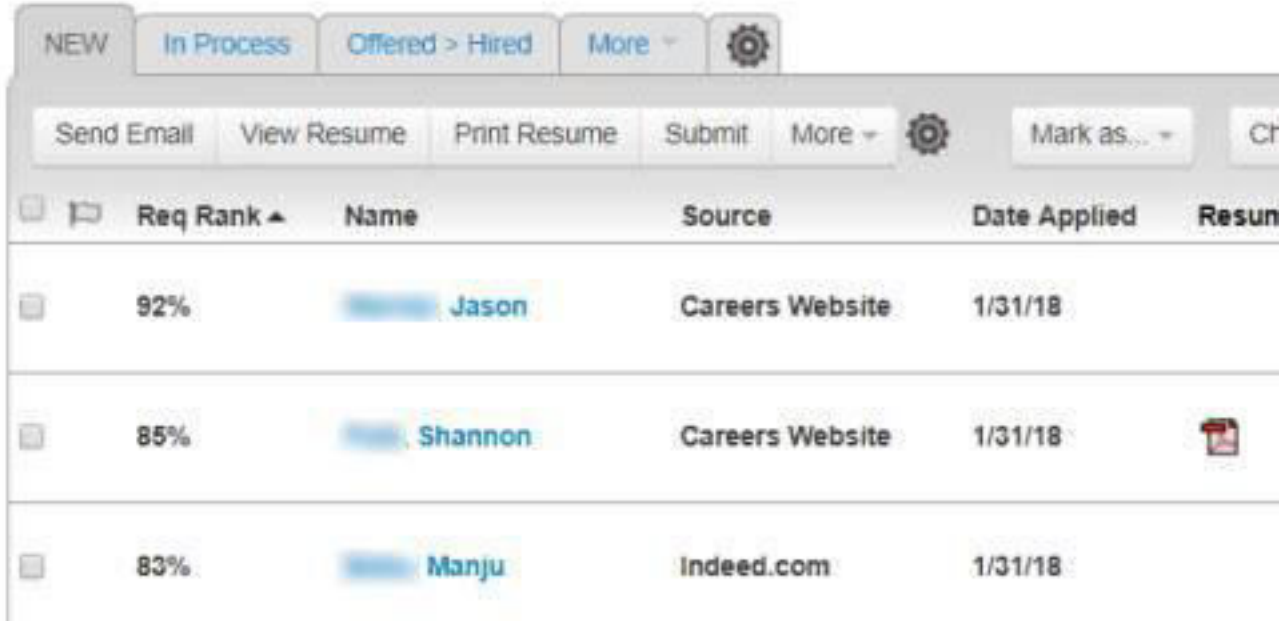
SAP SuccessFactors was born when SuccessFactors was acquired by SAP America in 2011. While technically a full human capital management (HCM) software, it's the 3rd most popular CV Filtering Tool among the Fortune 500.

iCIMS, the second biggest CV Filtering Tool by market share according to Datanyze, is a content generating machine. Fun fact: iCIMS is an acronym that stands for Internet Collaborative Information Management Systems.

Workday was named the leading vendor in Gartner's 2018 Magic Quadrant for Cloud HCM Suites Mid-market and Large Enterprises. According to Ongig, it was one of the fastest growing CV Filtering Tools by % growth in 2018.

Taleo, acquired by Oracle for $1.9 billion in 2012, is the leading recruiting software in the CV Filtering Tool category with 23% of market share according to Datanyze.

Demo for working of Taleo



The benefits of CV Filtering tool:

The key benefits include:

- Significantly speeds up the sifting process.Thus vast amounts of HR's time is saved.
- Keyword parsing and filtering reduces an otherwise time intensive task of sifting through applications

- Keyword matching and scoring helps find quality candidates
- Weeds out poor match candidates before the interview process begins
- Reduces ‘time-to-hire’ by as much as 40%
- Reduces ‘cost-per-hire’
- Reduces HR’s workload and thus allows HR to spend time on other activities
-

The disadvantages of CV Filtering tool:

The key disadvantages include:

- Many candidates are eliminated without the candidate receiving any reasons for rejection.Candidates may feel disrespected by this lack of explanation. This potential for 'candidate unfriendliness' may hurt the hiring company's image
- Automated criteria may damage diversity
- Human skill of choosing quality candidates is reduced to a number of keywords
- Software parsing technology has its limits e.g. known to struggle when applicants lay out resume in format not recognisable by software
- Software maybe poorly set up so keywords are too generic e.g. 'operations', 'communications' and 'management'. Here HR receives many unsuitable resumes and/or suitable resumes are filtered out
- Good candidates may be excluded from the running simply because they do not include certain keywords in their application

The most popular C.V Filtering tool

One thing we found is that software giant Oracle, itself a Fortune 500 company, continues to dominate this tool market. A handful of their offerings cropped up in this list, but Taleo stood above all with a massive 30.2% market share.

3. TECHNOLOGIES USED

Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.Python is a programming language that lets you work quickly and integrate systems more efficiently.

There are two major Python versions- Python 2 and Python 3. Both are quite different.It uses interpreter instead of compiler. # in the front of the line indicates that it is comment.It is strongly indented language.

4.1.1 Modules

A module is a file containing Python definitions and statements. A module can define functions, classes and variables. A module can also include runnable code. Grouping related code into a module makes the code easier to understand and use.

We can use any Python source file as a module by executing an import statement in some other Python source file. When interpreter encounters an import statement, it imports the module if the module is present in the search path.

A search path is a list of directories that the interpreter searches for importing a module.For example, to import the module calc.py, we need to put the following command at the top of the script, example: import math

print (math.sqrt(4))

The imaplib library:

- This module defines three classes, IMAP4, IMAP4_SSL and IMAP4_stream, which encapsulate a connection to an IMAP4 server and implement a large subset of the IMAP4rev1 client protocol as defined in RFC 2060. It is backward compatible with IMAP4 (RFC 1730) servers, but note that the STATUS command is not supported in IMAP4.

- This module defines three classes, IMAP4, IMAP4_SSL and IMAP4_stream, which encapsulate a connection to an IMAP4 server and implement a large subset of the IMAP4rev1 client protocol as defined in RFC 2060. It is backward compatible with IMAP4 (RFC 1730) servers, but note that the STATUS command is not supported in IMAP4.

- Three classes are provided by the imaplib module, IMAP4 is the base class:

  class imaplib.IMAP4(host='', port=IMAP4_PORT)

This class implements the actual IMAP4 protocol. The connection is created and protocol version (IMAP4 or IMAP4rev1) is determined when the instance is initialized. If host is not specified, '' (the local host) is used. If port is omitted, the standard IMAP4 port (143) is used.

When used like this, the IMAP4 LOGOUT command is issued automatically when the with statement exits.

.:>>> from imaplib import IMAP4

Three exceptions are defined as attributes of the IMAP4 class:

- exception IMAP4.error

  Exception raised on any errors. The reason for the exception is passed to the constructor as a string.

- exception IMAP4.abort

  IMAP4 server errors cause this exception to be raised. This is a sub-class of IMAP4.error. Note that closing the instance and instantiating a new one will usually allow recovery from this exception.

- exception IMAP4.readonly

  This exception is raised when a writable mailbox has its status changed by the server. This is a sub-class of IMAP4.error. Some other client now has write permission, and the mailbox will need to be re-opened to re-obtain write permission.

IMAP4 has many methods, few of the methods which were used in this project are:

- IMAP4.fetch(message_set, message_parts)

  Fetch (parts of) messages. message_parts should be a string of message part names enclosed within parentheses, eg: "(UID BODY[TEXT])". Returned data are tuples of message part envelope and data.

- IMAP4.login(user, password)

  Identify the client using a plaintext password. The password will be quoted.

- IMAP4.search(charset, criterion[, ...])

  Search mailbox for matching messages. charset may be None, in which case no CHARSET will be specified in the request to the server. The IMAP protocol requires that at least one criterion be specified; an exception will be raised when the server returns an error. charset must be None if the UTF8=ACCEPT capability was enabled using the enable() command.

- IMAP4.select(mailbox='INBOX', readonly=False)

Select a mailbox. Returned data is the count of messages in mailbox (EXISTS response). The default mailbox is 'INBOX'. If the readonly flag is set, modifications to the mailbox are not allowed.

The smtplib library:

The smtplib module defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon. For details of SMTP and ESMTP operation, consult RFC 821 (Simple Mail Transfer Protocol) and RFC 1869 (SMTP Service Extensions).

class smtplib.SMTP(host='', port=0, local_hostname=None, [timeout,]source_address=None)

An SMTP instance encapsulates an SMTP connection. It has methods that support a full repertoire of SMTP and ESMTP operations. If the optional host and port parameters are given, the SMTP connect() method is called with those parameters during initialisation. If speci-

fied, local_hostname is used as the FQDN of the local host in the HELO/EHLO command. Otherwise, the local hostname is found using socket.getfqdn(). If the connect() call returns anything other than a success code, an SMTPConnectError is raised.

The optional timeout parameter specifies a timeout in seconds for blocking operations like the connection attempt (if not specified, the global default

timeout setting will be used). If the timeout expires, socket.timeout is raised. The optional source_address parameter allows binding to some specific source address in a machine with multiple network interfaces, and/or to some specific source TCP port. It takes a 2-tuple (host, port), for the socket to bind to as its source address before connecting. If omitted (or if host or port

are '' and/or 0 respectively) the OS default behaviour will be used.

For normal use, you should only require the initialisation/connect, sendmail(), and SMTP.quit() methods. An example is included below.

The SMTP class supports the with statement. When used like this, the SMTP QUIT command is issued automatically when the with statement exits.

Eg: >>> from smtplib import SMTP

A nice selection of exceptions is defined as well:

- exception smtplib.SMTPException

  Subclass of OSError that is the base exception class for all the other exceptions provided by this module.

  Changed in version 3.4: SMTPException became subclass of OSError

- exception smtplib.SMTPServerDisconnected

  This exception is raised when the server unexpectedly disconnects, or when an attempt is made to use the SMTP instance before connecting it to a server.

- exception smtplib.SMTPResponseException

  Base class for all exceptions that include an SMTP error code. These exceptions are generated in some instances when the SMTP server returns an error code. The error code is stored in the smtp_code attribute of the error, and the smtp_error attribute is set to the error message.

- exception smtplib.SMTPSenderRefused

  Sender address refused. In addition to the attributes set by on

  all SMTPResponseException exceptions, this sets 'sender' to the string that the SMTP server refused.

- exception smtplib.SMTPRecipientsRefused

  All recipient addresses refused. The errors for each recipient are accessible through the attribute recipients, which is a dictionary of exactly the same sort as SMTP.sendmail() returns.

- exception smtplib.SMTPDataError

  The SMTP server refused to accept the message data.

- exception smtplib.SMTPConnectError

Error occurred during establishment of a connection with the server.

- exception smtplib.SMTPHeloError

  The server refused our HELO message.

- exception smtplib.SMTPNotSupportedError

  The command or option attempted is not supported by the server.

- exception smtplib.SMTPAuthenticationError

  SMTP authentication went wrong. Most probably the server didn't accept the username/password combination provided.

SMTP has many methods, few of the methods which were used in this project are:

- SMTP.ehlo(name='')

  Identify yourself to an ESMTP server using EHLO. The hostname argument defaults to the fully qualified domain name of the local host. Examine the response for ESMTP option and store them for use by has_extn(). Also sets several informational attributes: the message returned by the server is stored as

  the ehlo_resp attribute, does_esmtp is set to true or false depending on whether the server supports ESMTP, and esmtp_features will be a dictionary containing the names of the SMTP service extensions this server supports, and their parameters (if any).

  Unless you wish to use has_extn() before sending mail, it should not be necessary to call this method explicitly. It will be implicitly called by sendmail() when necessary.

- SMTP.starttls(keyfile=None, certfile=None, context=None)

  Put the SMTP connection in TLS (Transport Layer Security) mode. All SMTP commands that follow will be encrypted. You should then call ehlo() again.

  If keyfile and certfile are provided, they are used to create an ssl.SSLContext.

  Optional context parameter is an ssl.SSLContext object; This is an alternative to using a

keyfile and a certfile and if specified both keyfile and certfile should be None.

If there has been no previous EHLO or HELO command this session, this method tries ESMTP EHLO first.

SMTPHeloError

The server didn’t reply properly to the HELO greeting.

SMTPNotSupportedError

The server does not support the STARTTLS extension.

RuntimeError

SSL/TLS support is not available to your Python interpreter.

- SMTP.login(user, password, *, initial_response_ok=True)

  Log in on an SMTP server that requires authentication. The arguments are the username and the password to authenticate with. If there has been no

  previous EHLO or HELO command this session, this method tries ESMTP EHLO first. This method will return normally if the authentication was successful, or may raise the following exceptions:

  SMTPHeloError

  The server didn’t reply properly to the HELO greeting.

  SMTPAuthenticationError

  The server didn’t accept the username/password combination.

  SMTPNotSupportedError

  The AUTH command is not supported by the server.

SMTPException

No suitable authentication method was found.

Each of the authentication methods supported by smtplib are tried in turn if they are advertised as supported by the server. See auth() for a list of supported authentication methods. initial_response_ok is passed through to auth().

Optional keyword argument initial_response_ok specifies whether, for authentication methods that support it, an “initial response” as specified in RFC 4954 can be sent along with the AUTH command, rather than requiring a challenge/response.

- SMTP.sendmail(from_addr, to_addrs, msg, mail_options=(), rcpt_options=())

  Send mail. The required arguments are an RFC 822 from-address string, a list of RFC 822 to-address strings (a bare string will be treated as a list with 1 address), and a message string. The caller may pass a list of ESMTP options (such as 8bitmime) to be used in MAIL FROM commands as mail_options. ESMTP options (such as DSN commands) that should be used with all RCPT commands can be passed as rcpt_options. (If you need to use different ESMTP options to different recipients you have to use the low-level methods such as mail(), rcpt() and data() to send the message.

  Message may be a string containing characters in the ASCII range, or a byte string. A string is encoded to bytes using the ascii codec, and lone \r and \n characters are converted to \r\n characters. A byte string is not modified.

  If there has been no previous EHLO or HELO command this session, this method tries ESMTP EHLO first. If the server does ESMTP, message size and each of the specified options will be passed to it (if the option is in the feature set the server advertises).

If EHLO fails, HELO will be tried and ESMTP options suppressed.

This method will return normally if the mail is accepted for at least one recipient. Otherwise it will raise an exception. That is, if this method does not raise an exception, then someone should get your mail. If this method does not raise an exception, it returns a dictionary, with one entry for each recipient that was refused. Each entry contains a tuple of the SMTP error code and the accompanying error message sent by the server.

If SMTPUTF8 is included in mail_options, and the server supports it, from_addr and to_addrs may contain non-ASCII characters.

This method may raise the following exceptions:

SMTPRecipientsRefused

All recipients were refused. Nobody got the mail. The recipients attribute of the exception object is a dictionary with information about the refused recipients (like the one returned when at least one recipient was accepted).

SMTPHeloError

The server didn’t reply properly to the HELO greeting.

SMTPSenderRefused

The server didn’t accept the from_addr.

SMTPDataError

The server replied with an unexpected error code (other than a refusal of a recipient).

SMTPNotSupportedError

SMTPUTF8 was given in the mail_options but is not supported by the server.

Unless otherwise noted, the connection will be open even after an exception is raised.

- SMTP.quit()

  Terminate the SMTP session and close the connection. Return the result of the SMTP QUIT command.

  Low-level methods corresponding to the standard SMTP/ESMTP commands HELP, RSET, NOOP, MAIL, RCPT, and DATA are also supported. Normally these do not need to be called directly, so they are not documented here. For details, consult the module code.

The mammoth package

Mammoth is designed to convert .docx documents, such as those created by Microsoft Word, and convert them to HTML. Mammoth aims to produce simple and clean HTML by using semantic information in the document, and ignoring other details. For instance, Mammoth converts any paragraph with the style Heading 1 to h1elements, rather than attempting to exactly copy the styling (font, text size, colour, etc.) of the heading.

There's a large mismatch between the structure used by .docx and the structure of HTML, meaning that the conversion is unlikely to be perfect for more complicated documents. Mammoth works best if you only use styles to semantically mark up your document.

The following features are currently supported:

- Headings.
- Lists.
- Customisable mapping from your own docx styles to HTML. For instance, you could con-

vert WarningHeading to h1.warning by providing an appropriate style mapping.

- Tables. The formatting of the table itself, such as borders, is currently ignored, but the formatting of the text is treated the same as in the rest of the document.
- Footnotes and endnotes.
- Images.

- Bold, italics, underlines, strikethrough, superscript and subscript.
- Links.
- Line breaks.

The PyPDF2 package

A Pure-Python library built as a PDF toolkit. It is capable of:

- extracting document information (title, author, …)
- splitting documents page by page
- merging documents page by page
- cropping pages

By being Pure-Python, it should run on any Python platform without any dependencies on external libraries. It can also work entirely on StringIO objects rather than file streams, allowing for PDF manipulation in memory. It is therefore a useful tool for websites that manage or manipulate PDFs.

4. SYSTEM DESIGN

UML Diagrams

Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behaviour. To clarify a bit in details, dynamic behaviour means the behaviour of the system when it is running/operating.

So only static behaviour is not sufficient to model a system rather dynamic behaviour is more important than static behaviour. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

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

In brief, the purposes of use case diagrams can be as follows:

a. Used to gather requirements of a system.

b. Used to get an outside view of a system.

c. Identify external and internal factors influencing the system.

d. Show the interacting among the requirements are actors.

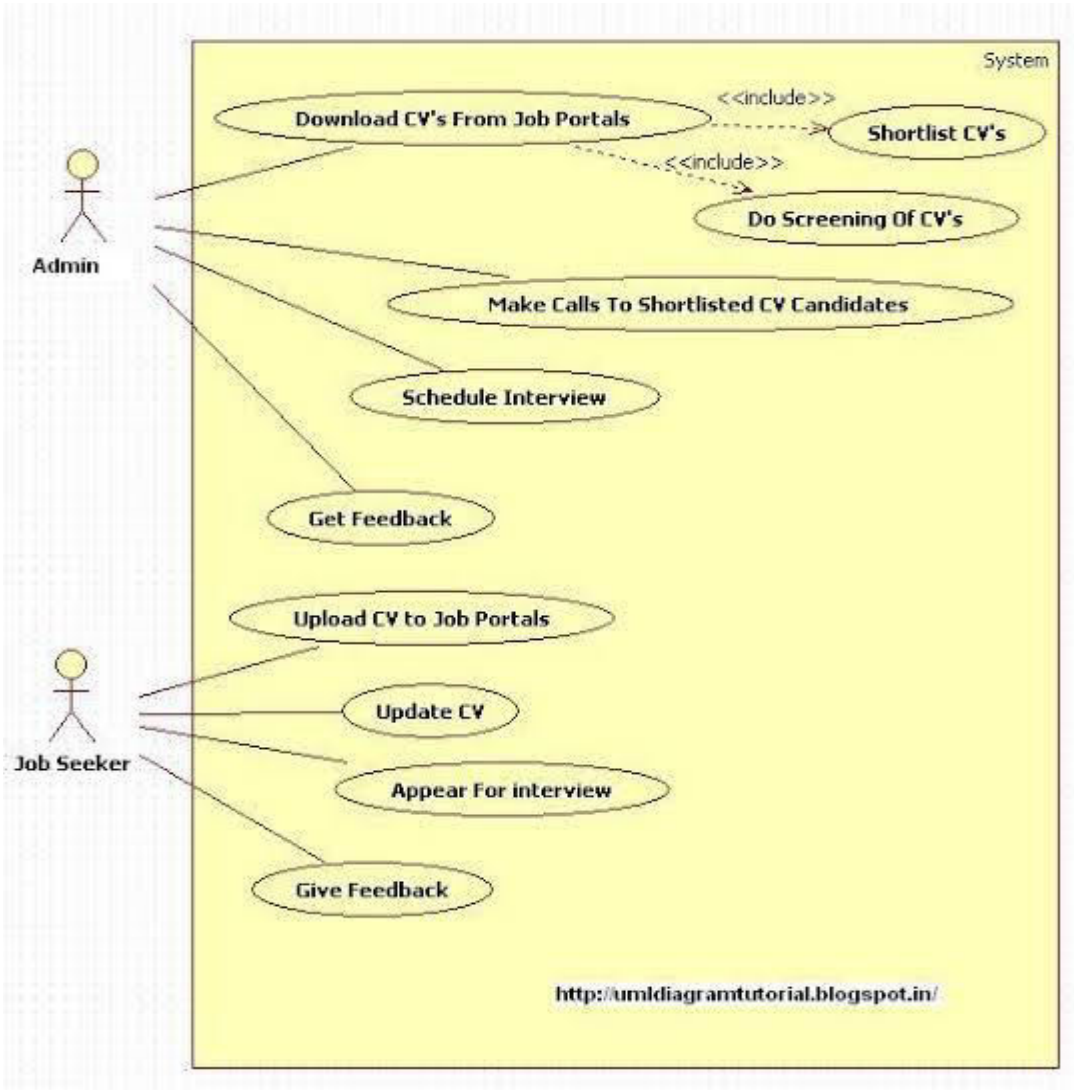


Figure11:Usecase Diagram for entire application functionality

5. IMPLEMENTATION

The success of the software product is determined only when it is successfully implemented according to the requirements. The analysis and the design of the proposed system provide a perfect platform to implement the idea using the specified technology in the desired environment.

Any software project is designed in modules and the project is said to be successfully implemented when each of the module is executed individually to obtain the expected result and also, when all the modules are integrated and run together without any errors.

Any project becomes successful only when it is user friendly and it is used by a major proportion of people.Python helps us to make such projects. Python was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.

Google blocks sign-in of a account through less secure apps like the terminal .So we need to turn the “Allow Less Secure apps” toggle ON in our gmail settings before using the program. Once it is turned on, user can use it any number of times.As we are using imap protocol, we also need to turn the ‘Enable IMAP’ toggle ON.The implementation of our project is made user friendly.

Flow Chart:

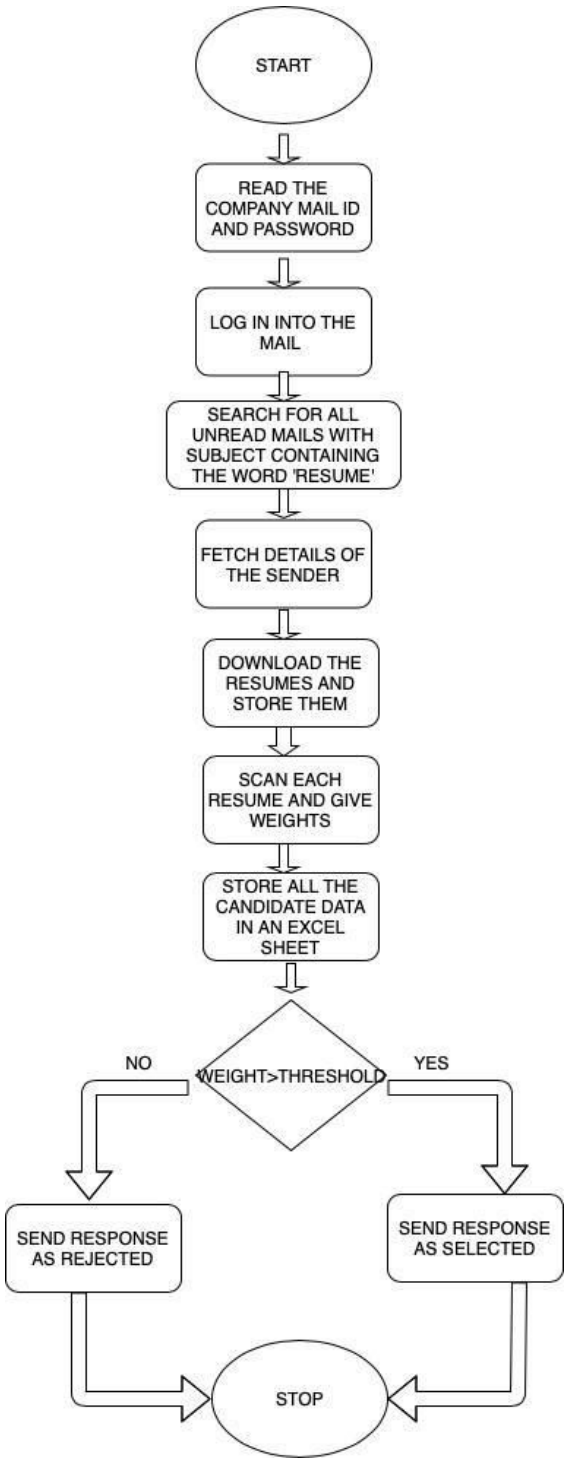


Fig.5.1-Flow Chart

Functions:

Function to search the inbox:

```
def inboxSearch():

    print('Searching for the resumes...\n\n')

    m = imaplib.IMAP4_SSL("imap.gmail.com")
    m.login(receiveId,receivePass) m.select(“inbox")
    resp, items = m.search(None,'(UNSEEN SUBJECT “resume")', )
```

This function searches for the mails with subject resume in the inbox of the company mail. It logs into the company’s mail id using login() function and selects the inbox using select() function

Deciding whether the applicant is selected or rejected:

```
import mammoth
with open("path", "rb") as docx_file:
    result = mammoth.extract_raw_text(docx_file)
    text = result.value

    text=text.lower()
    k=text.index('education')
    e=text[k+10:k+250]
    while True:

        t=e.index('.')
```

```
if e[t-1].isnumeric() and e[t+1].isnumeric():
            gpa=e[t-1:t+3]

            break
        e=e[t+1:]
    gpa=float(gpa)

    if gpa>7.00:
        w=5

    k=text.index('skills')
    skill=text[k+5:k+50]
    j=skill.count('java')
    p=skill.count('python')
    o=skill.count('operating system’)

w=w+j*3+p*2+o
    if 'projects' in text:
        w=w+1

    if 'experience' in text:
        w=w+2

if w>7.5:
    print('Selected')

else:
    print('rejected')
```

The above code is used to decide whether the applicant is eligible for the next round which can probably be a technical interview. The technical skills of the applicant like the programming languages and other technical knowledge is taken into consideration and given

weights based on the company's choice. Also other aspects like previous work experience, Cumulative Grade Point Average are taken into consideration in the selection process of the applicant.

Function to Save the applicants database in an excel sheet:

```
def saveInXl():

    print('Saving data in excel sheet...\n')
    wb=openpyxl.Workbook()
    sheet=wb.active

    sheet.title='resumes'
    sheet.cell(row=1,column=1).value='NAME'
    sheet.cell(row=1,column=2).value='PHONE NUMBER'
    sheet.cell(row=1,column=3).value='EMAIL ID'
    sheet.cell(row=1,column=4).value='DATE-TIME'
    sheet.cell(row=1,column=5).value='DECISION'



    sheet_row=2

    for downloaded_resume in received_from_data:
        sheet.cell(row=sheet_row,column=1).value=received_from_data[downloaded_resume][1]
        sheet.cell(row=sheet_row,column=2).value=received_from_data[downloaded_resume][4]
        sheet.cell(row=sheet_row,column=3).value=received_from_data[downloaded_resume][0]
        sheet.cell(row=sheet_row,column=4).value=received_from_data[downloaded_resume][2]
```

```
        sheet.cell(row=sheet_row,column=5).value=received_from_data[downloaded_resume][5]
        sheet_row+=1
```

This function after downloading the resumes from the mail into the local system extracts the information from the resumes like Name of the person, Phone number, Email ID, Date-time (at which the resume was sent) and Decision (whether the candidate is selected or rejected) and stores them in an excel sheet with rows and columns.

Function to send mail to the applicant:

```
def sendmail():

    print("Sending replies to candidates...\n ")
    smtpObj = smtplib.SMTP('smtp.gmail.com', 587)
    smtpObj.ehlo()

    smtpObj.starttls()
    useremail=receiveId
    password=receivePass
    smtpObj.login(useremail,password)

    wb=openpyxl.load_workbook(detach_dir+ '\\candidate_data.xlsx')
    sheet=wb.get_sheet_by_name('resumes')
```

```
    lastCol=5


    for r in range(2, 2+len(received_from_data)):
        decision=sheet.cell(row=r, column=lastCol).value
        name=sheet.cell(row=r, column=1).value
        senderemail=sheet.cell(row=r, column=3).value
```

```
if decision=='Yes':

        body = "Subject: SELECTED.\nDear %s,\n We are glad to inform you that you are selected for techincal interview." %(name)
    else:

        body = "Subject: rejected.\nDear %s,\n We are sorry to inform you that you are not selected for techincal interview." %(name)
    print('Sending email to %s...' % senderemail)

    sendmailStatus = smtpObj.sendmail(useremail, senderemail, body)
    if sendmailStatus != {}:

        print('There was a problem sending email to %s: %s' % (sendereemail,sendmailStatus))
smtpObj.quit()

print("Finished sending replies to candidates.\n\n")
```

The above function after storing the results in the excel sheet sends a mail to the applicant whether he/she is selected or rejected by using SMTP protocol.

6. TESTING AND RESULTS

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. In fact, testing is the one step in the software engineering process that could be viewed as destructive rather than constructive.

A strategy for software testing integrates software test case design methods into a well-planned series of steps that result in the successful construction of software. Testing is the set of activities that can be planned in advance and conducted systematically. The underlying motivation of program testing is to affirm software quality with methods that can economically and effectively apply to both strategic to both large and small-scale systems.

Testing Objectives

The main objective of performance testing is designed to test whether display is as expected and whether the webpage is functioning properly or not.

As the test results are gathered and evaluated they begin to give a qualitative indication of the reliability of the code. If proper output is not obtained, the overall quality of the code is questioned. If, on the other hand, all the results which are not successful, are encountered, and are easily modifiable, then the following conclusion can be made: The tests are inadequate as the requirements mentioned are not compatible. The testing includes:

- Checking whether the information is displayed or not.

- Checking whether all the players data is collected or not.

- Checking whether all the inputs are correctly taken or not.

- Verifying if all the pictures are displayed(if any) and none of the files are corrupted.

Output Screens

```
Sais-MacBook-Air:desktop k.sairuthwik$ python3 sample2.py
 checking emails  with subject 'RESUME ' and are unread..
 enter email id where the applicants have sent resumes
harinipopuri99@gmail.com
enter password
[redacted]
Specify the skills you are looking for in resumes. Enter all the skill seperated by
spaces.
example: python django java
python java c++
```

Fig. 6.1-Output 1

```
resumes-and-candidate-data\2019-10-24,18-19-14
Searching for the resumes...


[Pooja Ailani <poojaailani1974@gmail.com>] :Fwd: RESUME
[Kittu Kanchanapally <kittukanchanapally@gmail.com>] :Fwd: Resume for application
[Sai Ruthwik <sairuthwik511@gmail.com>] :My resume
[kavya sumanika <ksumanika@gmail.com>] :RESUME
[Nihaal Bandapalli <nihaalsp@gmail.com>] :Resume
Finished downloading resumes.


Scanning all the resumes...
```

Fig. 6.2-Output 2

```
Finished scanning all the resumes.


Saving data in excel sheet...

Finished saving data in excel sheet.


Sending replies to candidates...

Sending email to poojaailani1974@gmail.com...
Sending email to kittukanchanapally@gmail.com...
Sending email to sairuthwik511@gmail.com...
Sending email to ksumanika@gmail.com...
Sending email to nihaalsp@gmail.com...
Finished sending replies to candidates.
```

Fig. 6.3-Output 3

The terminal where the program is compiled and run

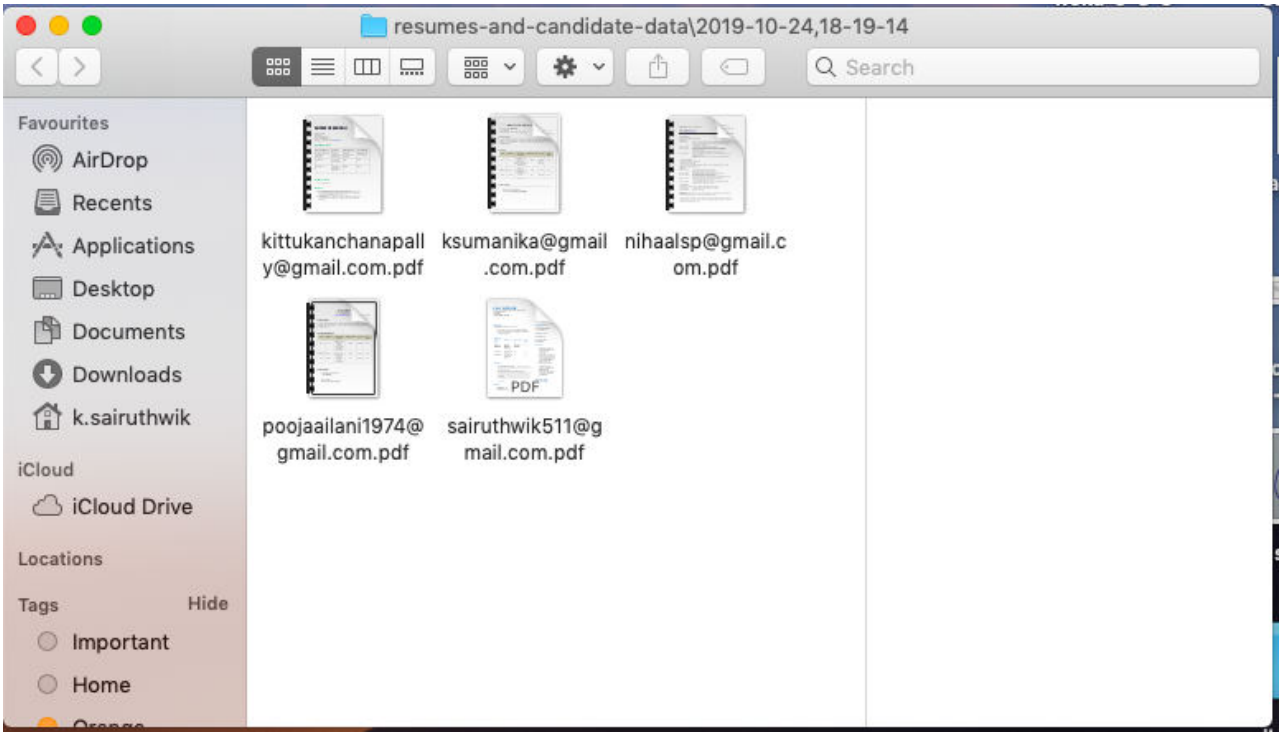


Fig.6.4-Output 2

The local folder with the downloaded resumes.

NAME	PHONE NUMBER	EMAIL ID	DATE-TIME	DECISION
Pooja Ailani	8639841276	poojaailani1974@gmail.com	Thu, 24 Oct 2019 18:16:55 +0530	Yes
Kittu Kanchanapally	-	kittukanchanapally@gmail.com	Thu, 24 Oct 2019 18:14:37 +0530	No
Sai Ruthwik	-	sairuthwik511@gmail.com	Thu, 24 Oct 2019 18:14:13 +0530	Yes
kavya sumanika	9133494492	ksumanika@gmail.com	Thu, 24 Oct 2019 18:13:18 +0530	No
Nihaal Bandapalli	-	nihaalsp@gmail.com	Thu, 24 Oct 2019 18:12:16 +0530	No

Fig. 6.5-Output 3

The excel sheet with the candidate's data

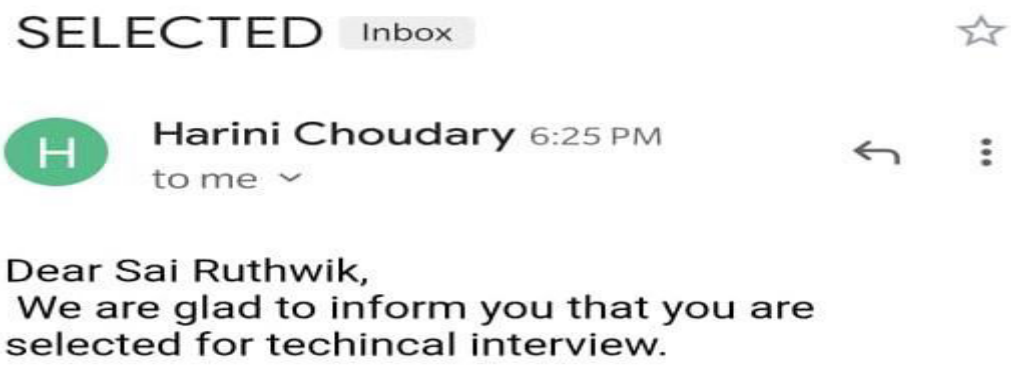


Fig. 6.6-Output 4

The response mail sent to a selected candidate.

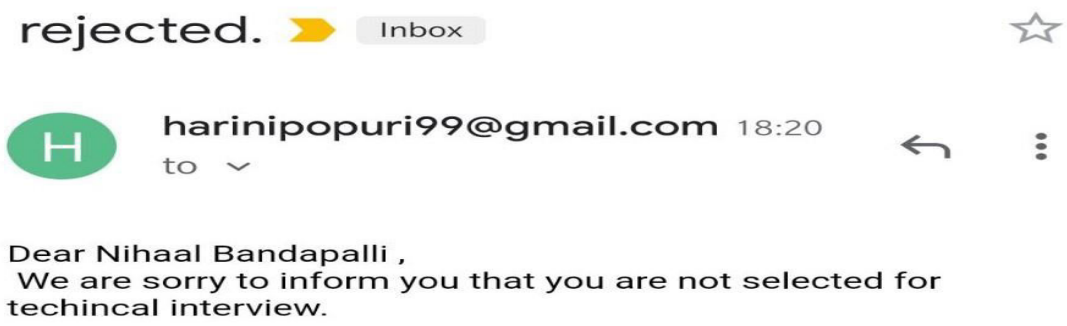


Fig. 6.7-Output 5

The response mail sent to a rejected candidate.

7. CONCLUSION AND FUTURE SCOPE

This is a very useful project as it reduces the burden on the human resource department of the company.It is very simple as the user needs to give the company mail id and password only, the rest of the work is done by the program only.

The project has a lot of future scope. If the emerging technologies like neural networks, deep learning are implemented to scan the resumes and rank them, then the project can be established as a end product.

8. BIBLIOGRAPHY

[1] https://docs.python.org/3/library/imaplib.html

[2] https://docs.python.org/3/library/smtplib.html

[3] https://nevonprojects.com/personality-prediction-system-through-cv-analysis/