**Recruitment Management Tool**

**ABSTRACT**

This project is a simple example of how a recruitment-management-tool is used by the HR managers to scan a set of resumes and filter out a certain number of resumes that match the required skills.

The project is done in C language using text files to store and retrieve all the necessary information.

The required skills for a particular job is initially stored in a file and the individual resumes are stored in separate files. The program scans the resumes for matching skills from the required skill list.

The criteria for shortlisting a resume is, at least 50% of the skills must be found in the resume.

The shortlisted resumes are put into another file for the HR to reference.

**REQUIREMENTS**

**Functional Requirements:**

* The program must retrieve the skill list from skills.txt
* The program must retrieve the resume names from candidates.txt
* The program must properly access the individual resume files and retrieve the individual skills of each candidates.
* The program must count the number of matching skills from each resume.
* The resumes with 50% or more matching skill list must be shortlisted and stored into another file.

**System Requirements:**

* IDE used to code: Eclipse Mars.
* C compiler like gcc. (MinGW used to test the project)
* Compiler support for following inbuilt functions: strcmpi(), strcspn(), strcat(), fopen(), fgets, fseek(), fgetc(), fputc().

**DESIGN**

START

Read List of skill one by one from the file mentioned in RESUMENAME

Count=0

If end of file is encountered

Read next line from candidates.txt and store into variable RESUMENAME

Read all skills from skills.txt and stores into A

**YES**

**NO**

If end of RESUMENAME file is encountered

Count=Count+1

If a skill in RESUMENAME matches with skills in skills.txt

**YES**

**NO**

**YES**

**B**

**C**

**A**

**B**

**C**

**C**

If count>(0.50\*contents of skills.txt)

**A**

Copy the Contents of RESUMENAME file into the shortlist.txt file

**NO YES**

**TESTING**

**Unit Testing:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test case No. | Test Description | Test Data | Status | Comments |
| **1.** | There are no matching skills or less than 50% matching skills in all of the resumes | Skills.txt has 5 skills and candidates.txt has 3 filenames and none of the files have more than 2 matching skills | **PASS**  The shortlist file is empty at the end of the operation as none of the candidate is eligible | Normal operation of the program |
| **2.** | 2 resumes have 50% or more matching skills among 5 resumes that exactly match the names in skills.txt | Candidates.txt has 5 filenames among which 2 have more than 50% of skills in the skills.txt file | **PASS**  The shortlist file has resume details of 2 candidates at the end of the operation | Normal operation of the program |
| **3.** | 2 resumes have 50% or more matching skills among 5 resume that match the names in skills.txt but the names are in different case. | Candidates.txt has 5 filenames among which 2 have more than 50% of skills in the skills.txt file | **PASS**  The shortlist file has resume details of 2 candidates at the end of the operation | This test case proves the program is designed such that the case of the skill names are insensitive.  Eg: PRograMMing will be matched with programming |
| **4.** | 2 resumes have 50% or more matching skills among 5 resume that match the names in skills.txt but the 3rd resume has one matching skill repeated 5 times. | Skills.txt has 5 skills and candidates.txt has 5 filenames | **PASS**  The shortlist file has resume details of 2 candidates at the end of the operation. The 3rd resume has rejected as it does not qualify. | This test case proves that the program is designed such that the repeated skill will not deter the performance and will deliver the correct result |
| **5.** | All the resumes have 50% or more skills matching with the required skills | Skills.txt has 5 skills and candidates.txt has 3 filenames and all of the files have more than 2 matching skills | **PASS**  The shortlist file contains all the resume details of files in candidates.txt at the end of the operation | Normal operation of the program |

**Defects:**

* As the storage medium is files instead of database, the retrieval and storage method is not very efficient.
* Since there is no GUI, the HR might not be familiar with the application.
* There is no retrieval option for individual resumes.

**CONCLUSION**

The recruitment management tool is an efficient method for obtaining the shortlisted resumes of candidates that are eligible for the required job description.

The implementation is effective if the HR has huge number of resumes to process and it saves lot of time which could otherwise be used for something else.

**REFERENCES**

* <https://stackoverflow.com>
* <https://www.ibm.com/support/knowledgecenter/en>