MEDIATING PLATFORM FOR JOB SEEKERS AND TALENT ACQUISITION MANAGERS

UCS2201 – Fundamentals and Practice of Software Development

A PROJECT REPORT

Submitted By

Janani Hariharakrishnan , 3122225001046

K Hishitha, 3122225001040

Madhumitha E ,3122225001068

Department of Computer Science and Engineering

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

Kalavakkam – 603110

July 2023

Sri Sivasubramaniya Nadar College of Engineering

(An Autonomous Institution, Affiliated to Anna University)

BONAFIDE CERTIFICATE

Certified that this project report titled "**MEDIATING PLATFORM FOR JOB SEEKERS AND TALENT ACQUISITION MANAGERS**" is the bonafide work of "Janani Hariharakrishnan (3122225001046), K Hishitha, (3122225001040), Madhumitha E (3122225001068)" who carried out the project work in the UCS2201 – Fundamentals and Practice of Software Development during the academic year 2022-23.

Internal Examiner External Examiner

Date:

TABLE OF CONTENTS

Abstract

Our project aims to develop an innovative platform that facilitates seamless interaction between job seekers and talent acquisition managers. This platform acts as a central hub, enabling both parties to fulfill their respective needs effectively. Job seekers can take advantage of advanced search and matching capabilities to explore job opportunities that align with their profiles. By utilizing the platform's cutting-edge algorithms and extensive database, job seekers can find positions that best match their skills and qualifications. Similarly, talent acquisition managers have access to powerful search functionality that helps them efficiently identify qualified candidates who meet their specific job requirements. This platform streamlines the recruitment process and fosters a more efficient and productive hiring experience for talent acquisition managers by simplifying candidate selection and connection.

Problem Statement

Our objective is to develop a sophisticated mediating platform that facilitates seamless interaction between job seekers and talent acquisition managers. This platform serves as a central hub for both parties to effectively meet their respective needs:

i. Job seekers can leverage the platform to explore open positions that align with their profiles. By utilizing advanced search and matching capabilities, they can identify job opportunities that best suit their skills and qualifications.

ii. Talent acquisition managers are empowered to search for deserving applicants who possess the necessary qualifications to meet their specific job requirements. Through the platform's comprehensive database and powerful search functionality, they can efficiently identify and connect with
potential candidates.

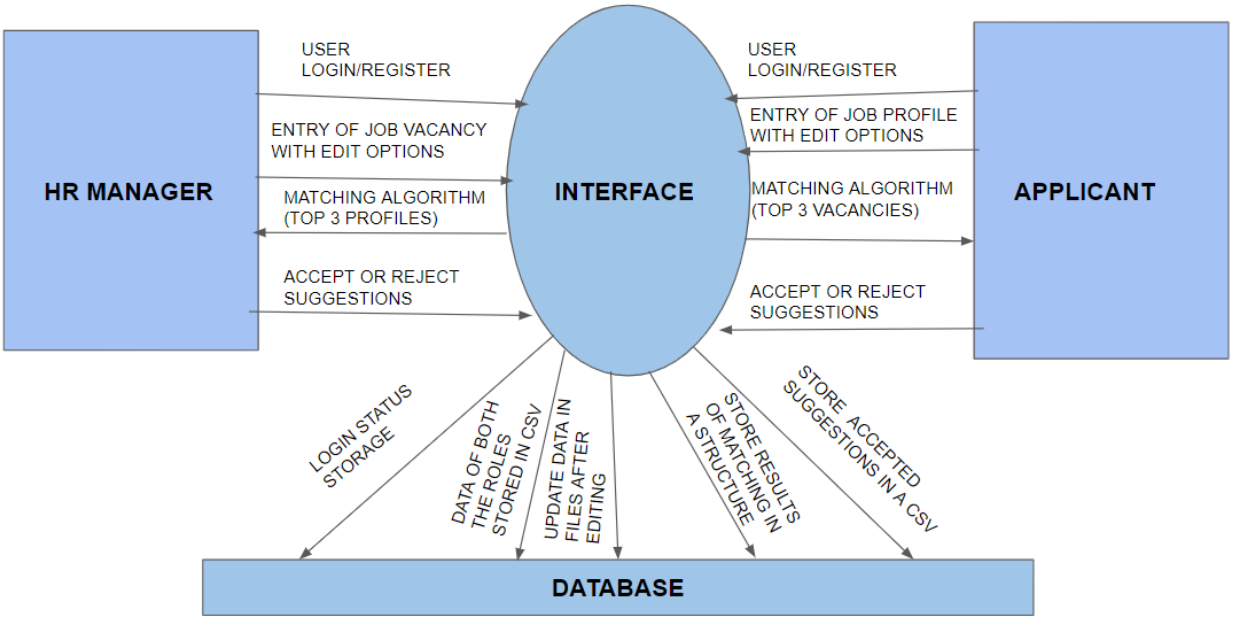
Exploring the problem further beyond the given statement

By providing a robust and user-friendly interface, our mediating platform streamlines the job search and talent acquisition processes, enhancing efficiency and effectiveness for both job seekers and talent acquisition managers.

- Develop a secure login page with robust authentication mechanisms and industry-standard data protection for user verification.
- Implement user-friendly profiles for job seekers to upload comprehensive information and update their details seamlessly.
- Utilize a sophisticated matching algorithm that considers job seeker preferences and suggests relevant job positions.
- Empower job seekers to accept or decline suggested job matches, providing control and flexibility in their search.
- Provide talent acquisition managers with a streamlined interface to enter vacancy details and use a matching algorithm to suggest potential candidates, facilitating informed decision-making.

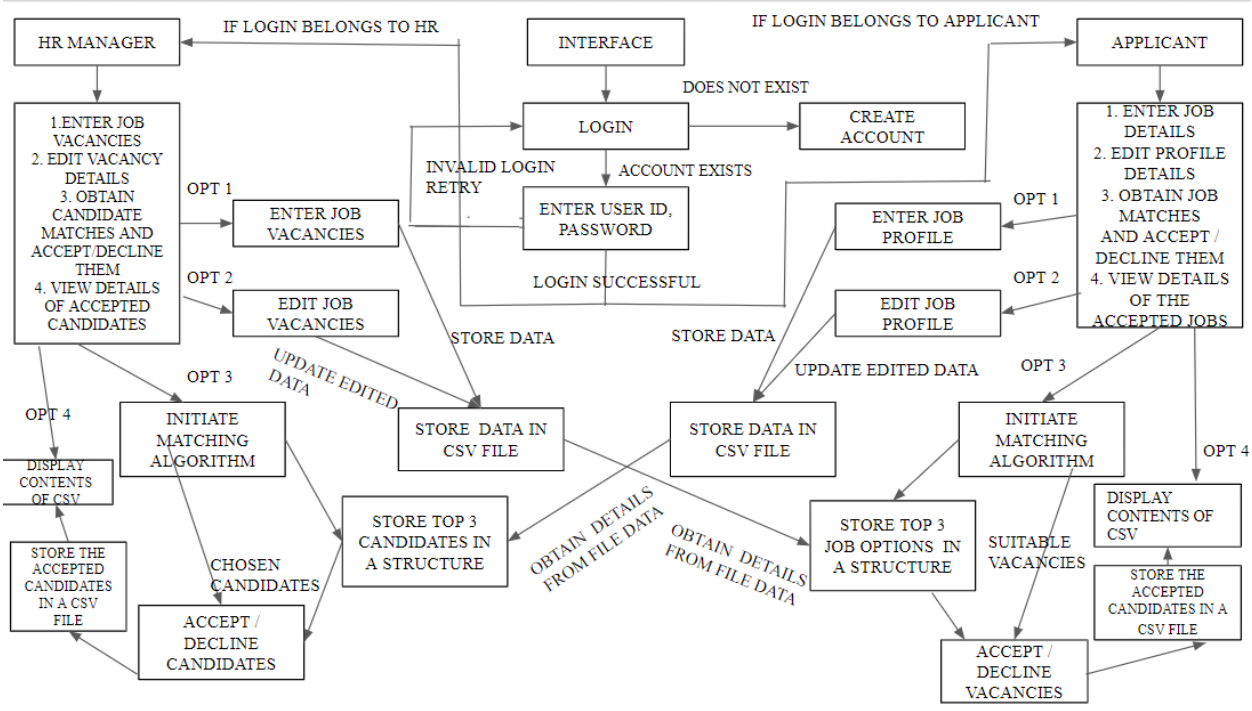
Problem Analysis through Diagrams:

DFD Level 0:



- The project comprises three core systems: the HR Manager, the Applicant, and the Interface. Users, whether HR Managers or Applicants, access the Interface by providing their login credentials or registering with their relevant details.
- Within the Interface, users can input job vacancy information, including any edited details as necessary. If a user wishes to find suitable candidates for a specific job vacancy or job profile, the Interface utilizes a matching algorithm to present the top three matches.
- Users are then given the option to accept or reject any number of these matches. The Interface efficiently manages the storage and updating of login details, data, profiles, top three matches, and accepted matches within the database.

Extended DFD Level 1:



The user interface offers the option to either log in with existing credentials or create a new account. If the user already has an account, they are prompted to provide their login ID and password. The system verifies these details by checking against the existing information stored in a secure CSV file. In the case of a new user, they are guided to create an account by providing their personal details. The entered information is securely stored in the CSV file, and a unique 6-digit login ID is assigned to the user. Upon successful verification of login credentials, the user is directed to their respective account, which is automatically recognized as either an HR manager or an applicant based on their stored details.

If the user is an HR manager, they have the following options:

1. Enter Job Vacancy: The HR manager can input job vacancy details, which are saved separately in the CSV file based on whether they are mandatory or desirable requirements.
2. Edit Job Details: The HR manager can update job vacancy information, and the revised data is saved in the existing CSV file, replacing the previous details.
3. Profile Matching and accept or reject suggested candidates : The system matches job profiles with applicant resumes. Compulsory details are compared, and applicants whose resumes do not meet these requirements are skipped. Aggregate scores are calculated based on the weights assigned by the HR manager for each field. The top three matches are stored in a structure and are displayed to the HR manager. Now, the HR manager has a choice to select the candidates among them. The selected candidates are then stored in a CSV file along with their details.
4. View details of selected candidates: The HR manager can view the details of the selected candidates. These details have been retrieved from the CSV file in which they were stored.

If the user is a job applicant, the system automatically recognizes their role, and they have the following options:

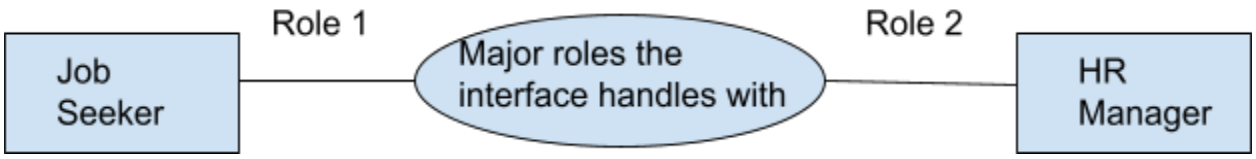
1. Enter Job Profile: The applicant can input their job profile details, which are saved in the CSV file according to whether they are compulsory or desirable fields.
2. Edit Profile Details: The applicant can modify their job profile information, and the updated data is saved in the existing CSV file.
3. Profile Matching and accept/reject suggested jobs: The system matches the applicant's job profile with existing job vacancies. Compulsory requirements are checked, and vacancies that do not match the compulsory requirements are skipped. A matching algorithm calculates aggregate scores based on the weights assigned by the applicant for each field. The top three matches with the highest aggregate scores remain in the structure, while others are removed. The top matches are then displayed to the applicant. Ang the

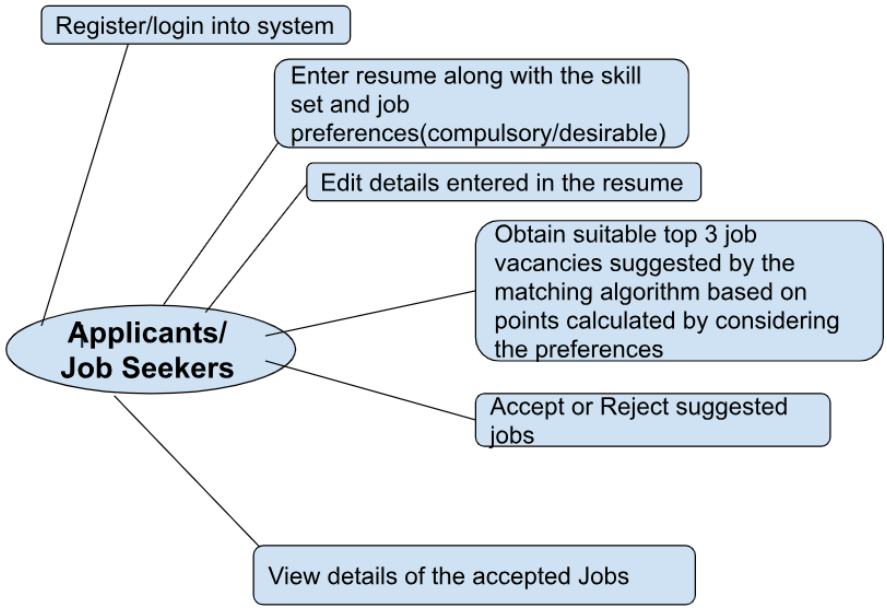
applicant can choose those Jobs that he wishes to apply from among the suggested jobs. These accepted jobs are stored in a CSV file along with job details.

4. View details of accepted jobs: The Applicant can view the details of the accepted jobs. The Jobs and the job details are retrieved from the CSV file.

This user interface allows HR managers and job applicants to efficiently manage their tasks, ensuring accurate storage and retrieval of information while facilitating seamless interaction with the system.

Use case Diagram:

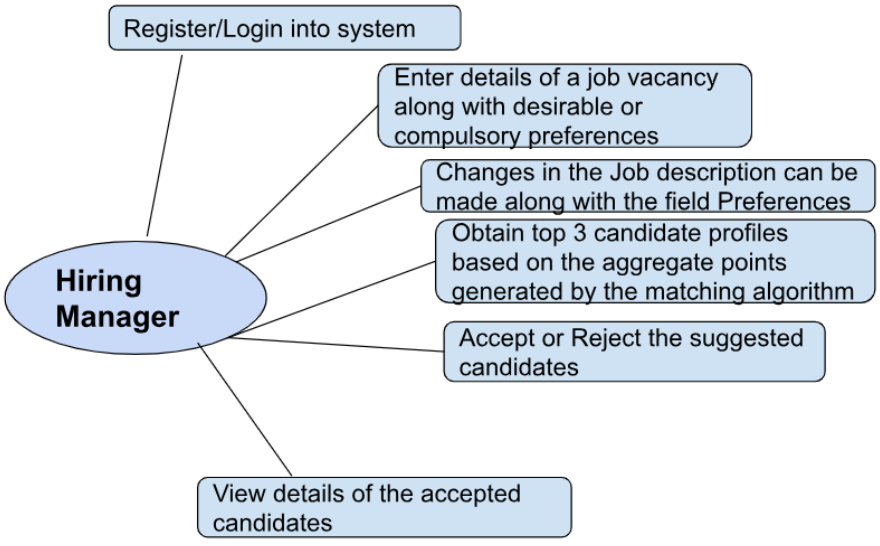




The interface in the project serves two key roles: Job Seeker and HR Manager.

For the Job Seeker/Applicant, the interface provides the following features:

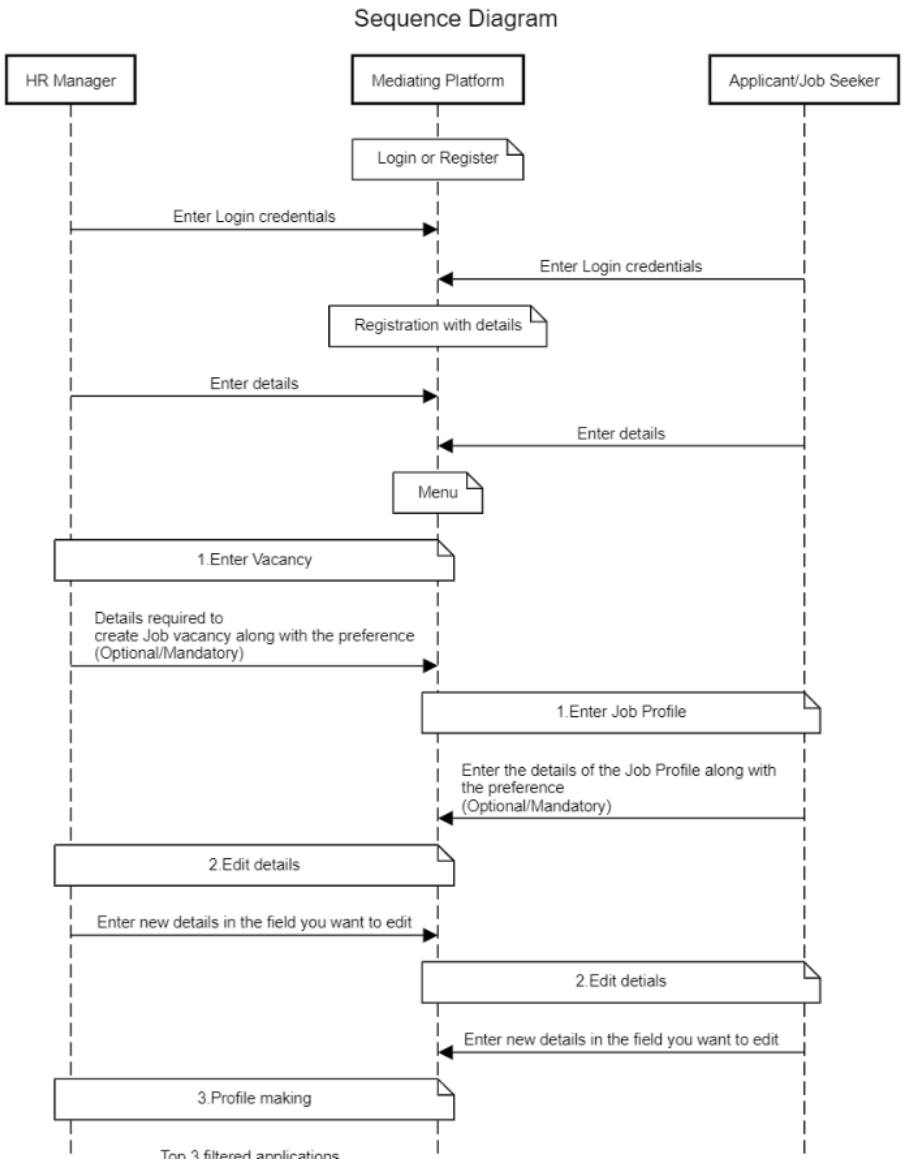
Authentication: The Job Seeker has the option to either log in or register into the system.
Resume Management: The Job Seeker can upload and maintain their resume, including skill sets and job preferences (mandatory or optional choices). They also have the ability to edit their resume details.
Job Matching: Using the matching algorithm, the system calculates a score based on the Job Seeker's preferences and generates a list of the top three best-matched job opportunities. The Job Seeker can review and either apply to or reject these job offers
Accept/Reject: The Job seeker can accept or reject jobs from among the suggested jobs
View details of the accepted jobs: The applicant can view the jobs and the job details that he has accepted.

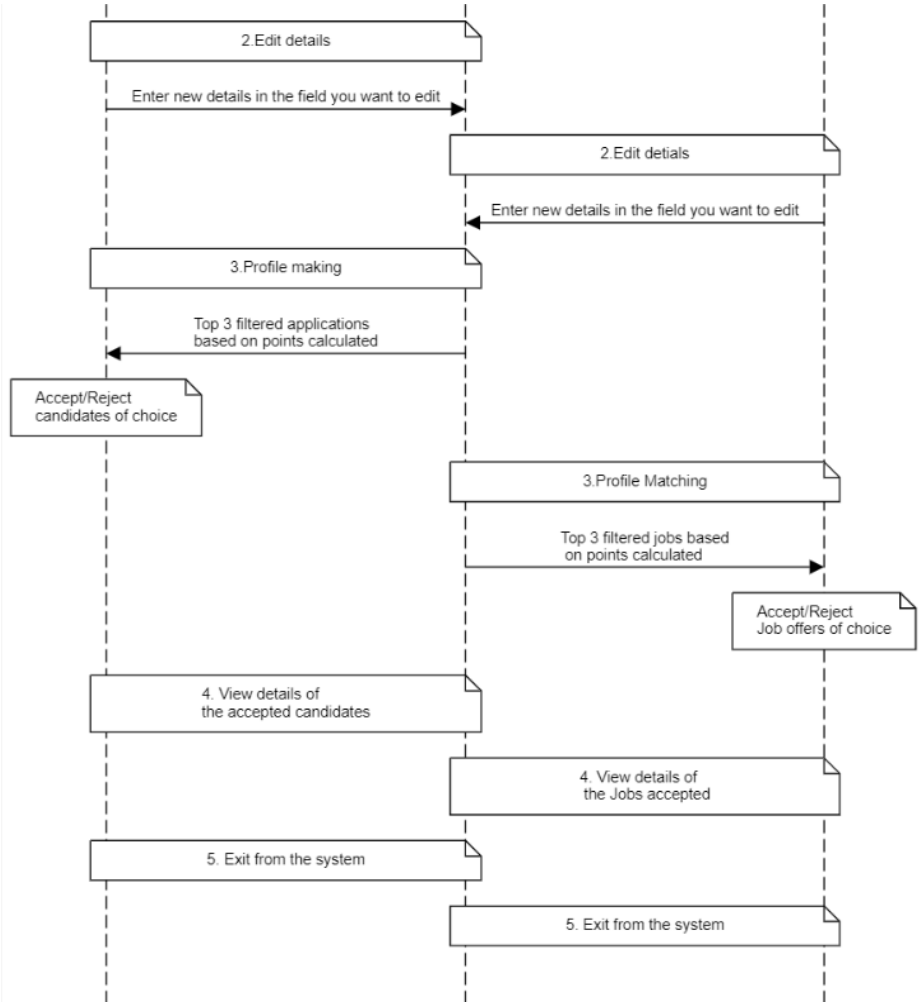


For the HR Manager, the interface provides the following functionalities:

Authentication: The HR Manager can log in or register into the system.
Job Vacancy Creation: The HR Manager can create vacancies by entering job details and requirements. He has the ability to modify the details of created job vacancies as needed.
Profile matching and accept/reject them : If desired, the HR Manager can utilize the system to identify the top three most suitable applicants for a given vacancy. The interface displays these applicants, and the HR Manager can choose to accept or reject them.
View details of the accepted profiles: Those job profiles that the HR manager has accepted can be displayed to the HR manager along with the job details.

Sequence Diagram:




2.Edit details
Enter new details in the field you want to edit
2.Edit detials
Enter new details in the field you want to edit
3.Profile making
Top 3 filtered applications
based on points calculated
Accept/Reject
candidates of choice
3.Profile Matching
Top 3 filtered jobs based
on points calculated
Accept/Reject
Job offers of choice
4. View details of
the accepted candidates
4. View details of
the Jobs accepted
5. Exit from the system
5. Exit from the system

HR MANAGER:

Login or Register into the system:

- The HR Manager can login with their credentials into the system.

Registration with details:

- If the HR manager does not have an account, he can register into the system by giving his details like email id, phone number and many other details to the interface.

Enter Vacancy:

- Hiring managers enters details of the Job vacancies along with the compulsory and optional requirements of the job like, including factors like salary, night shift availability,Transportation(shuttle/bus service), Number of paid leaves per year, Flexible work schedule and availability of work from home facility ,Insurance coverage for family ,Stock options or any other long-term incentives and other relevant factors.

Edit details:

- Allow updation of the entered details of the entered vacancy.
- The HR manager can make modifications to the job vacancy details they have previously entered. To initiate the editing process, the HR manager will need to provide the firm name and the manager name for verification purposes. Once authenticated, they can enter the field they want to update. The entered details are then updated in the data repository.

Profile Making:

- The HR manager can find the top three suitable applicants for a specific job vacancy they have created.
- The sophisticated matching algorithm(Compare the applicant's qualifications, experience, skills, and preferences with the requirements

specified by the HR for the job.)
- Calculate points based on the preferences set by the HR manager
- Display the most suitable applicants accordingly.

Accept or Decline the profile:

- The HR manager has the authority to accept or reject the job applications based on their own criteria and preferences from among the top 3 applicants.

View details of the accepted candidates:

- The HR manager can view the details of those candidates that he has accepted.

Exit from the system

- The HR manager can exit out of the complete system

APPLICANT OR JOB SEEKER:

Login or Resistor into the system:

- The Applicant can login with their credentials into the system.

Registration with details:

- If the Applicant does not have an account, he can register by providing his details like email id, phone number etc.

Enter Job Profile:

- The applicant to upload their job profile details. The applicant is required to provide information such as qualifications, skills, experience, and other relevant details. Additionally, they have to indicate whether each entered detail is mandatory or optional, along with assigning weightage points. All the provided information is securely stored in the data repository.

Edit details:

- The applicant can modify their existing job profile details. This feature enables them to update their qualifications, skills, or any other pertinent information. The revised data will be promptly updated and saved in the data repository.

Profile Matching:

- The applicant matches their job profile with existing job vacancies. Leveraging a sophisticated matching algorithm, the system presents the top three best-matched job vacancies based on the applicant's preferences and criteria.

Accept or Decline Job Offers:

- The applicant can provide and respond to job offers generated by the matching algorithm. They have the discretion to accept or reject the offered job based on their own considerations.

View details of the accepted Jobs:

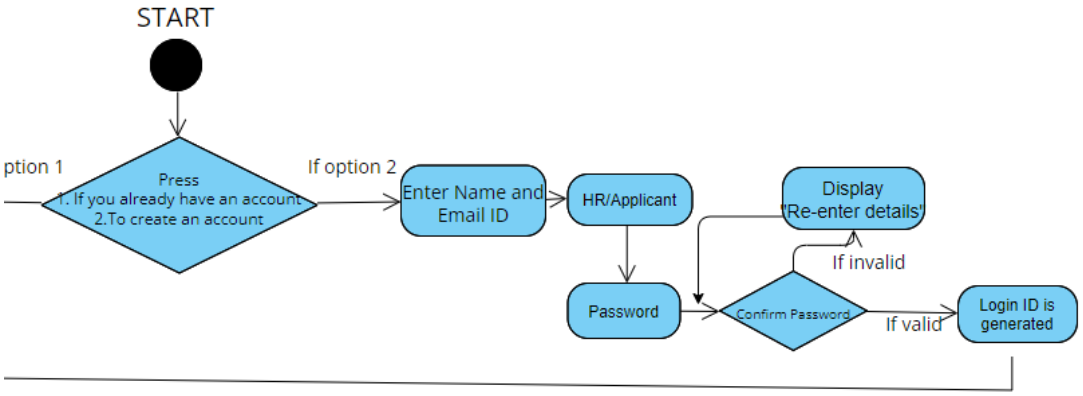
- The Applicant can view the jobs that he chooses to accept. The job details are also displayed along with it.

Exit from the system

- The applicant can exit out of the complete system

Activity Diagram:

1. Create an Account:



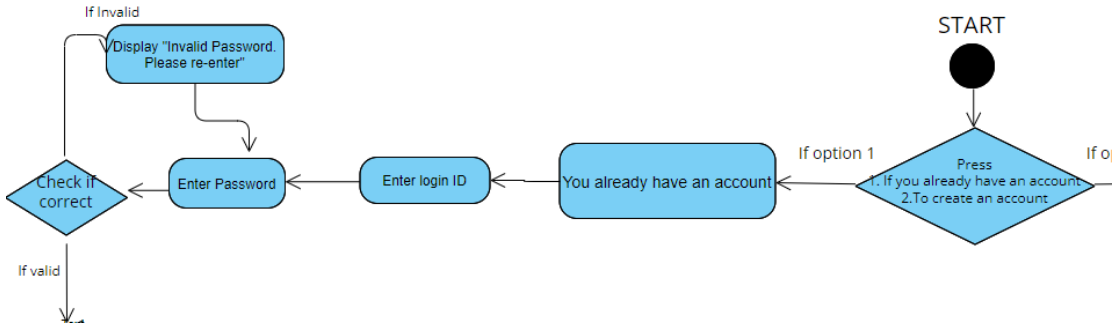
At the program's outset, users are presented with the option to either log in to their existing account or create a new one if they do not possess one. By selecting option 2, they can proceed with creating a new account.

This is a pivotal component of the login module, allowing users without an account to register. During the registration process, users are required to provide essential information, including their full name, email address, and specify whether they identify as an HR Manager or a Job applicant.

Subsequently, users are prompted to set a password and confirm it to ensure accuracy. A thorough validation check is performed to verify the correct confirmation of the password. Upon successful confirmation, a six-digit login ID is automatically generated using a secure random number generator algorithm.

This accurately describes the login module's functionality and highlights the importance of user account creation with precise information and secure password practices.

2. Login to Existing Account:

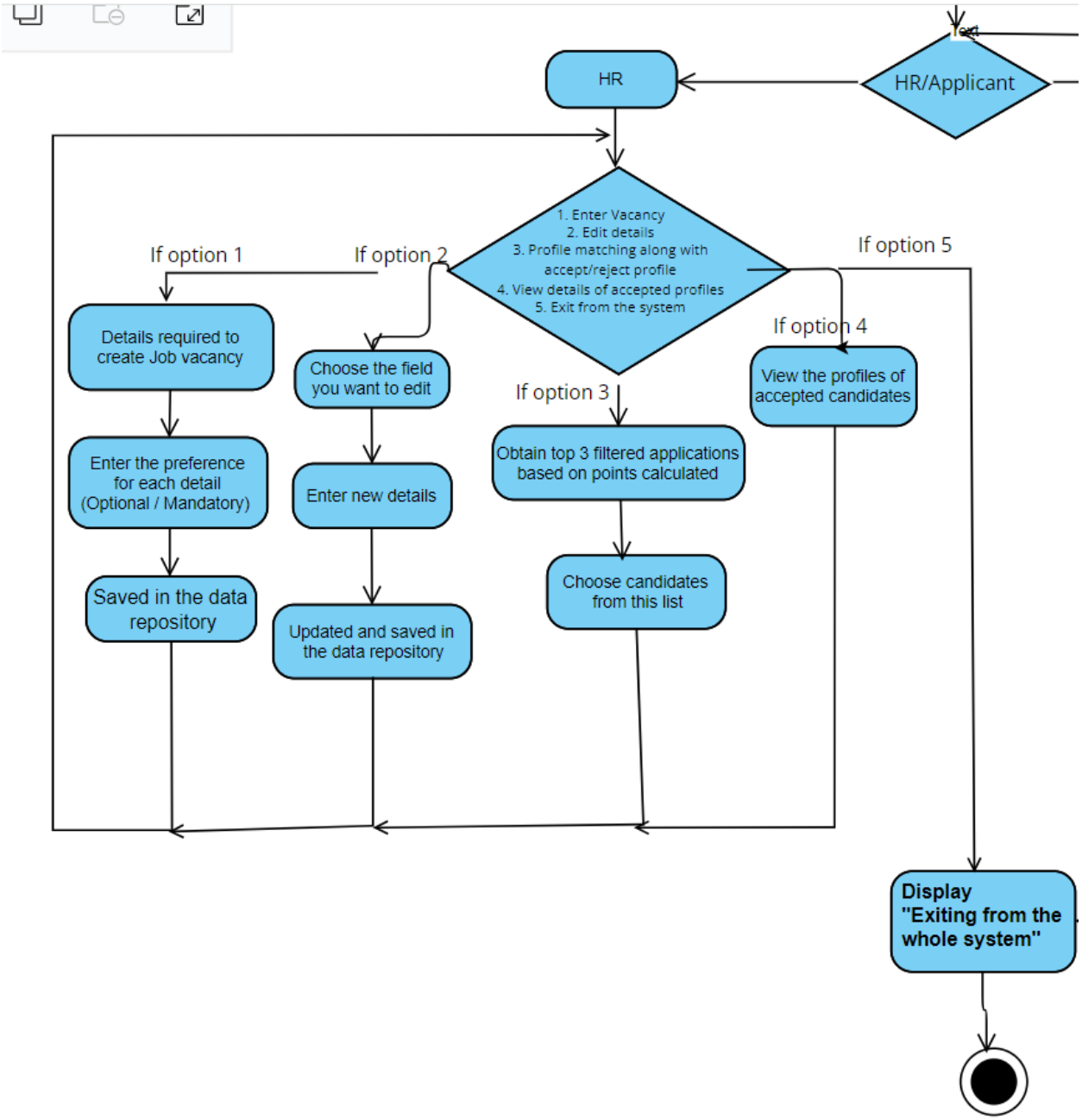


For users who already have an existing account, they are required to select option 1 to proceed with the login process. During login, users are prompted to enter their login credentials, which include their unique six-digit login ID and corresponding password.

To ensure secure access, the system validates the provided login credentials. If the credentials are accurate and match the stored information, the user is granted access to their respective account. However, if the login credentials are incorrect, the user is prompted to re-enter the details until the correct credentials are provided.

This professional content emphasizes the login process for users with existing accounts, highlighting the need for accurate login credentials and the iterative nature of re-entering details until the correct information is provided.

3. HR Manager:

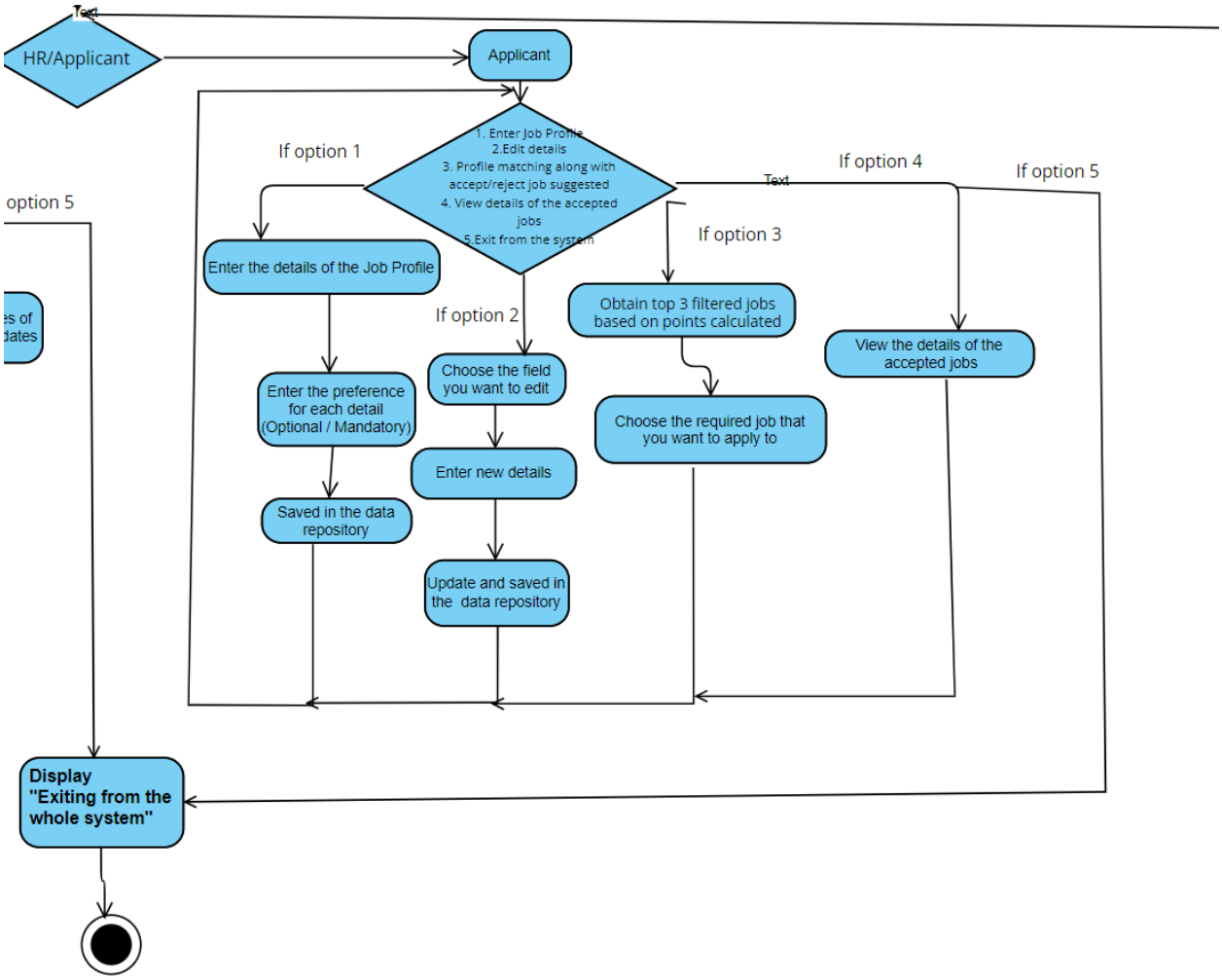


Once the user is logged into their account, their role as either an HR manager or an applicant is recognized based on saved information. The HR manager is presented with a main menu offering four major tasks they can perform:

1. Enter Job Vacancy: This option allows the HR manager to create a new job vacancy. They will be prompted to provide details such as job position, required educational qualifications, educational institution, work experience, aggregate score, salary range, and more. For each information field, the HR manager can specify whether it is a compulsory or desirable requirement and assign a corresponding weightage. All the entered details are securely saved in a data repository.
2. Edit Job Details: By selecting this option, the HR manager can make modifications to the job vacancy details they have previously entered. To initiate the editing process, the HR manager will need to provide the firm name and the manager name for verification purposes. Once authenticated, they can update the necessary information, and the changes will be saved in the data repository.
3. Profile Matching: This option enables the HR manager to find the top three suitable applicants for a specific job vacancy they have created. The sophisticated matching algorithm calculates points based on the preferences set by the HR manager and displays the most suitable applicants accordingly.Now, the HR manager has the authority to accept or reject the job applications based on their own criteria and preferences.
4. View the details of accepted candidates: This option enables the HR manager to take a look at the candidates and the candidate details which he had selected after matching the profile. He will have to enter the name of the firm and the manager name to look through the details.
5. Exit from the system: By choosing this option the user can come out of the whole system.

By utilizing this professional interface, HR managers can efficiently perform their tasks, such as creating job vacancies, editing details, finding suitable applicants, and managing the acceptance or rejection process and view details.

4. Job Seeker/Applicant:



Once Logged in, the user's role is automatically identified based on the information stored in the database. If the user is an applicant, they will be presented with a main menu offering four primary tasks:

1. Enter Job Profile: This option allows the applicant to input their job profile details. The applicant is required to provide information such as qualifications, skills, experience, and other relevant details. Additionally, they have to indicate whether each entered detail is mandatory or optional, along with assigning weightage points. All the provided information is securely stored in the data repository.
2. Edit Details: By selecting this option, the applicant can modify their existing job profile details. This feature enables them to update their qualifications, skills, or any other pertinent information. The revised data will be promptly updated and saved in the data repository.
3. Profile Matching: This option empowers the applicant to match their job profile with existing job vacancies. Leveraging a sophisticated matching algorithm, the system presents the top three best-matched job vacancies based on the applicant's preferences and criteria. Now, the applicant can review and respond to job offers generated by the matching algorithm. They have the discretion to accept or reject the offered job based on their own considerations.
4. View details of the accepted jobs: By choosing this option, the applicant can view the details of the jobs that he has accepted. He will have to enter the name of the firm and the manager name to take a view of these.
5. Exit from the system: By choosing this option the user can come out of the whole system.

By utilizing this professional interface, applicants can efficiently manage their job profiles, make necessary edits, leverage profile matching capabilities, accepting,viewing and exiting from the system.

How Is TheData Organized:

```
struct hr_suggestions { … } hr_suggestions[100], temp;

struct ap_suggestions { … } ap_suggestions[100], temp1;

struct hr_comp { … } hr_comp[100];

struct hr_des { … } hr_des[100];

struct ap_comp { … } ap_comp[100];

struct ap_des { … } ap_des[100];
```

```
FILE *file =
    fopen("hr_details_comp.csv", "a"); // Open th
FILE *file1 =
    fopen("hr_details_des.csv", "a"); // Open the
FILE *file2 =
    fopen("ap_details_comp.csv", "a"); // Open th
FILE *file3 = fopen("ap_details_des.csv", "a");
```

```
FILE *file =
    fopen("user_details.csv", "a");
```

```
FILE *file4 = fopen("hr_ranks.csv", "r");
```

```
FILE *file = fopen("login1.csv", "r");
```

```
FILE *file5 = fopen("ap_ranks.csv", "r");
```

Array of Structures :

- hr_suggestions:Stores the details of accepted suggestions by the HR Managers
- ap_suggestions:Stores the details of accepted suggestions by the applicants
- hr_comp:Stores the compulsory details of the HR Managers
- hr_des:Stores the desirable details of the HR Managers
- ap_comp:Stores the compulsory details of the Job Seekers
- ap_des:Stores the desirable details of the Job Seekers

CSV Files :

hr_ranks:Stores the details of accepted suggestions by the HR Managers

- ap_ranks:Stores the details of accepted suggestions by the applicants
- hr_details_comp:Stores the compulsory details of the HR Managers
- hr_details_des:Stores the desirable details of the HR Managers
- ap_details_comp:Stores the compulsory details of the Job Seekers
- ap_details_des:Stores the desirable details of the Job Seekers

Arrays are also used to store input details in the form of strings.

Using structures to store data before adding it to a file offers several advantages. Structures allow for the organization of related data fields, creating a logical and well-structured representation. This grouping facilitates the management and manipulation of complex entities.

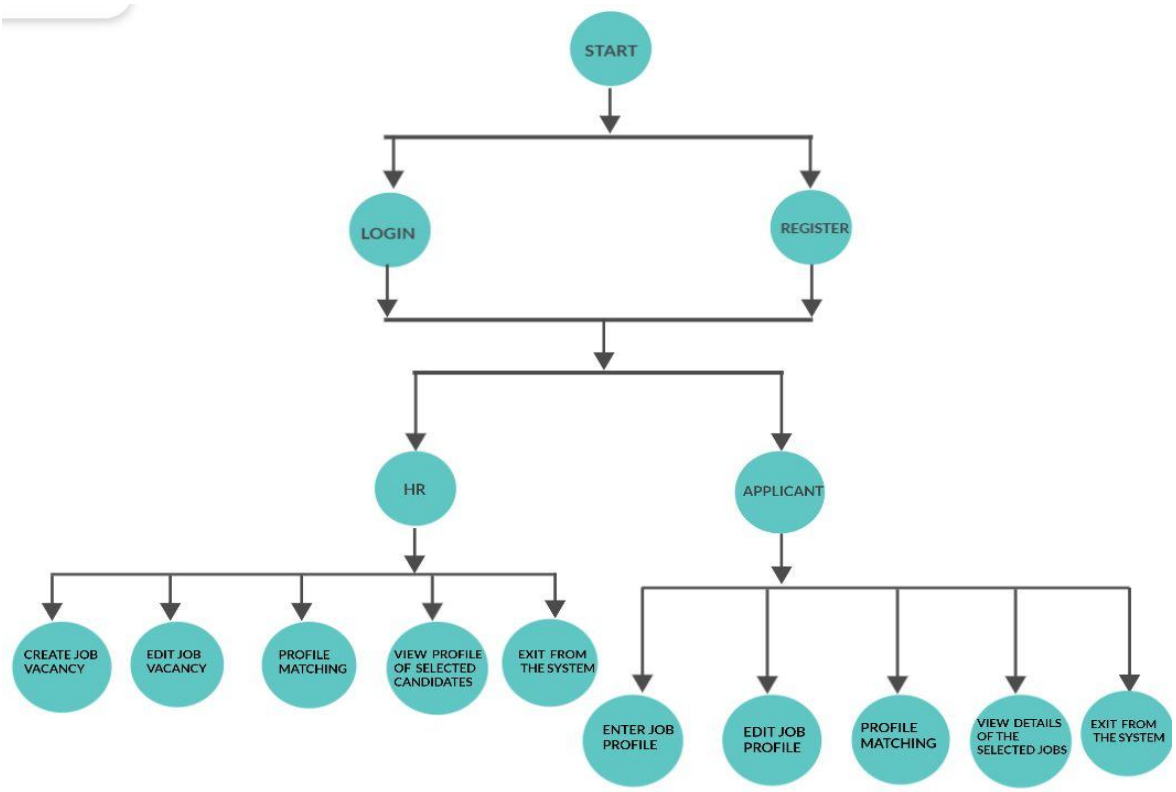
Structures also enforce data integrity by specifying data types and constraints, ensuring that the stored data adheres to the defined structure. This practice improves code readability and maintainability, providing a clear and organized representation of the data.

When writing the data to a file, you can create an instance of the structure, populate its fields, and use file I/O functions like fwrite() or fprintf() to write the entire structure, resulting in consistent and organized storage.

LIBRARIES USED IN THE CODE:

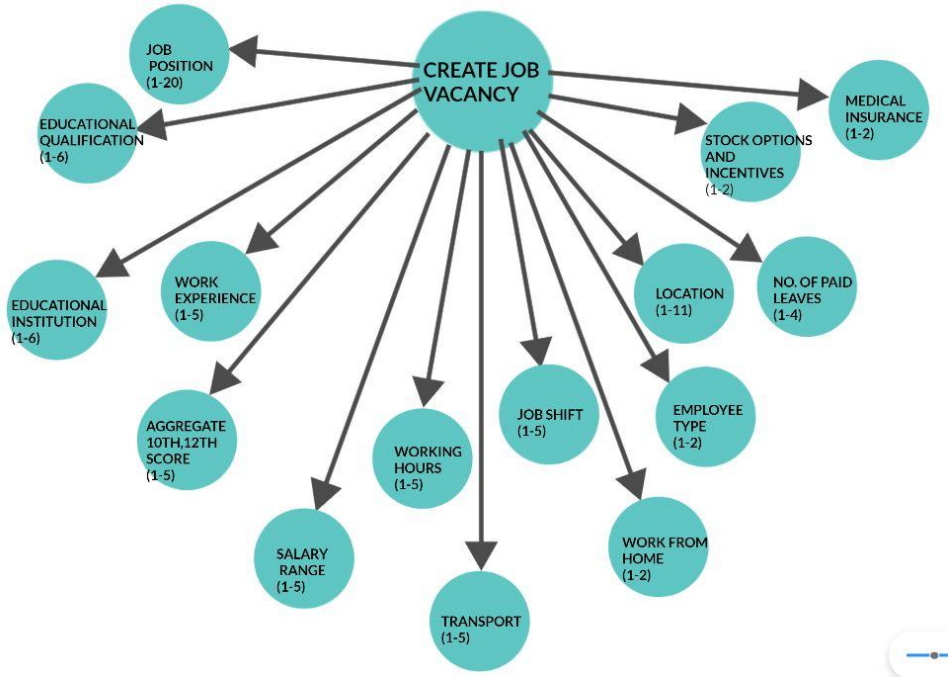
- <stdbool.h>: used to define the bool data type, which is used in the function is_unique() and the checkCredentials() function. It allows the use of boolean values (true and false) for returning the result of certain conditions.
- <stdio.h>: This library is used for standard input/output operations. (printf(), scanf(), and fgets())
- <stdlib.h>: used for various standard library functions such as exit() and srand(). In the provided code, it is used in the is_unique() function to exit the program if an error occurs while opening the file.
- <string.h>: used for string manipulation functions(functions like strcmp(), strcpy(), strtok(), strcspn(), and strlen())
- <time.h>: used to seed the random number generator in the generate_login_id() function using srand(time(NULL)).
- <unistd.h>: This library is used for various system call functions.

User Interface Design- Menu driven



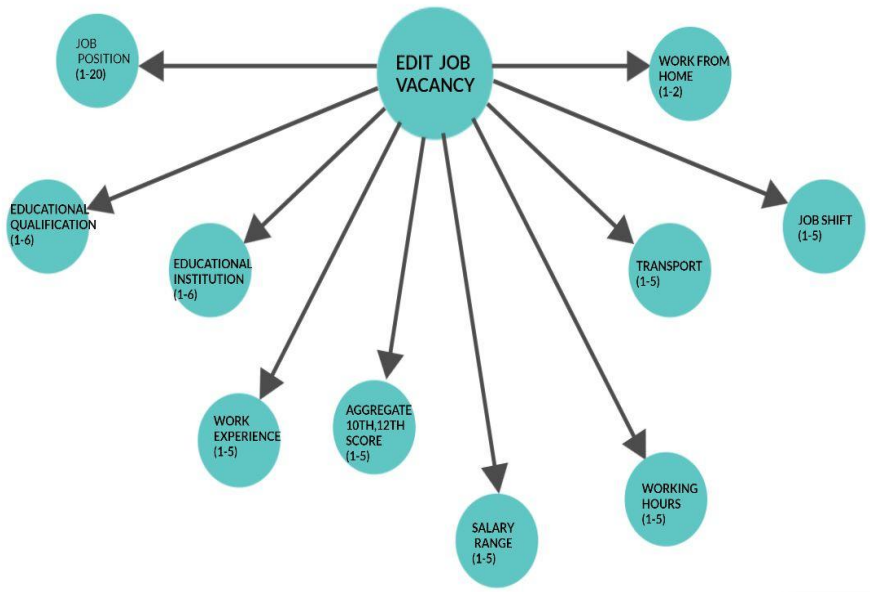
HR MANAGER:

CREATE JOB VACANCY:



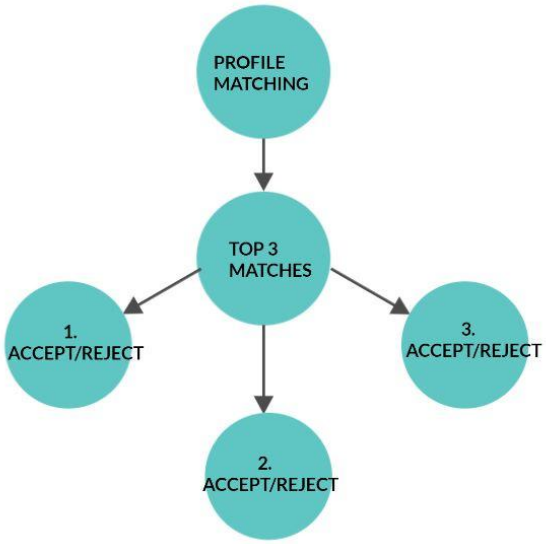
To create a new job vacancy. They will be prompted to provide details such as job position(with 1-20 option), required educational qualifications(with 1-6 options), educational institution(1-6), work experience(1-5), aggregate score(1-5), salary range(1-5),working hours(1-5),transport(1-5),job shift(1-5),work from home(1-2),employee type(1-2),location(1-11),number of paid leaves(1-4),stock options and incentives(1-2),medical insurance(1-2). For each information field, the HR manager can specify whether it is a compulsory or desirable requirement and assign a corresponding weightage. All the entered details are securely saved in a data repository.

EDIT JOB VACANCY:



The HR manager can make modifications to the job vacancy details they have previously entered. To initiate the editing process, the HR manager will need to provide the firm name and the manager name for verification purposes. Once authenticated, they can update the necessary information, and the changes will be saved in the data repository.

PROFILE MATCHING:

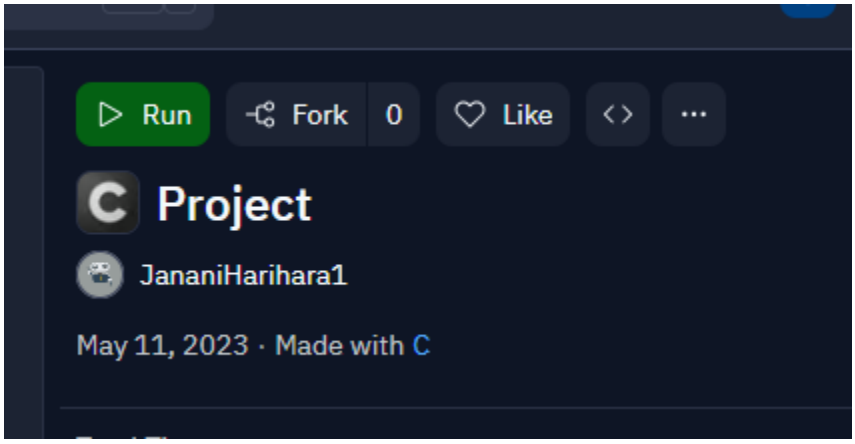


The HR manager to find the top three suitable applicants for a specific job vacancy they have created. The sophisticated matching algorithm calculates points based on the preferences set by the HR manager and displays the most suitable applicants accordingly.Now, the HR manager has the authority to accept or reject the job applications based on their own criteria and preferences.

VIEW ACCEPTED CANDIDATES:

The applicant can view the accepted candidates details of the selected jobs that he has accepted. He will have to enter the name of the firm , the applicant name , career field and login id to take a view of accepted suggestions.

PLATFORM USED FOR CODE DEVELOPMENT-REPLIT



- Replit enables multiple team members to work together on the same project simultaneously.
- Provides a comprehensive IDE tailored for coding in C, offering essential tools for development.
- Supports Git integration, facilitating effective version control and project management.
- It operates in the cloud, eliminating the need for local installations and allowing easy access, sharing, and deployment of the project.

Explaining the Matching Algorithm:

The algorithm follows these steps:

1. Iterate through each applicant's profile:

    - Retrieve relevant details from the applicant's file, such as qualifications, experience, skills, and preferences.
    - Extract job perks information from the applicant's preferences imputed during Job Profile entry, including factors like salary, night shift availability,Transportation(shuttle/bus service),Number of paid

leaves per year,Flexible work schedule and availability of work from home facility ,Insurance coverage for family ,Stock options or any other long-term incentives and other relevant factors.

- Prompt the applicant to provide information on whether each job perk is compulsory or desirable for them during the Profile entry stage itself.

2. Perform matching and scoring:

- Compare the applicant's qualifications, experience, skills, and preferences with the requirements specified by the HR for the job.
- Check if the compulsory factors mentioned by the HR match the applicant's preferences. If any compulsory factor does not match, move on to the next applicant.
- 5 x (points for the particular field for a compulsory factor)and 2.5 x (points for the particular field for a desirable factor) are allotted prior during the JOB vacancy entry stage.
- If all compulsory factors match, compute a total point inclusive of both the factors for the applicant based on the matching criteria and store it along with the matched applicant's details in a data structure (e.g., an array of structures).

3. Display Top 3 Matches:

- Once all applicants have been evaluated, sort the list of applicants based on their average points in descending order.
- Display the details of the top three applicants with the highest average points.

During the data entry , the points and preference are taken from the user.

```
        printf("\nStock options and incentives");
        printf("\n1.Provided");
        printf("\n2.Not Provided");
        printf("\nChoose from the above 2 options(1-2):");
        int stock;
        scanf("%d", &stock);
        printf("\n\nHow many points is alloted this particular
field (out "
               "of            10):");
        scanf("%f", &point);
        printf("\n Is this field compulsorily or"
               "optional? Enter 1 for compulsory or 0 for
optional");
        int stock_scr;
        scanf("%d", &stock_scr);
        if (stock_scr == 1) {
          strcpy(field, "Stock options and incentives\0");
          fprintf(file, "%s,%d,%f,", field, stock, 5 * point);
          fprintf(file1, "%d,%d,%d,", 0, 0, 0);
        } else {
```

```
  } else {
    strcpy(field, "Stock options and incentives\0");
    fprintf(file1, "%s,%d,%f,", field, stock, 2.5 * point);
    fprintf(file, "%d,%d,%d,", 0, 0, 0);
  }
```

Finding the entered profile and iterating through applicants to match compulsory details first

```
for (int y = 0; y < i; y = y + 2) {
  // to find hr details from compulsory file;
  if (strcmp(hr_comp[y].firm, name10) == 0 &&
      strcmp(hr_comp[y].name, hr10) == 0 &&
      strcmp(hr_comp[y].row2, job10) == 0) {
    flag = 1;
    // matching details of the compulsory file
    for (int y1 = 0; y1 < i; y1++) {
      sum = 0;
      sum1 = 0;
      if ((strcmp(hr_comp[y].row4, ap_comp[y1].row4) == 0) &&
          (strcmp(hr_comp[y].row1, ap_comp[y1].row1) == 0) &&
          (strcmp(hr_comp[y].row2, ap_comp[y1].row2) == 0) &&
          (strcmp(hr_comp[y].row5, ap_comp[y1].row5) == 0) &&
          (strcmp(hr_comp[y].row7, ap_comp[y1].row7) == 0) &&
          (strcmp(hr_comp[y].row8, ap_comp[y1].row8) == 0) &&
          (strcmp(hr_comp[y].row10, ap_comp[y1].row10) == 0) &&
          (strcmp(hr_comp[y].row11, ap_comp[y1].row11) == 0) &&
          (strcmp(hr_comp[y].row13, ap_comp[y1].row13) == 0) &&
          (strcmp(hr_comp[y].row14, ap_comp[y1].row14) == 0) &&
          (strcmp(hr_comp[y].row16, ap_comp[y1].row16) == 0) &&
          (strcmp(hr_comp[y].row17, ap_comp[y1].row17) == 0) &&
          (strcmp(hr_comp[y].row19, ap_comp[y1].row19) == 0) &&
```

Once all the compulsory factors have matched,
Fields like name of APPLICANT , email of APPLICANT and their compulsory matched sum is stored.

```
hr_suggestions[k].compsum = sum;
strcpy(hr_suggestions[k].name, ap_comp[y1].name);
char email[30];
strcpy(email, ap_comp[y1].name);
strcat(email, "@hotmail.com");
strcpy(hr_suggestions[k].email, email);
if ((strcmp(hr_des[y].row1, ap_des[y1].row1) == 0) &&
    (strcmp(hr_des[y].row2, ap_des[y1].row2) == 0)) {
  sum1 = sum1 + atof(hr_des[y].row3);
}
if ((strcmp(hr_des[y].row4, ap_des[y1].row4) == 0) &&
    (strcmp(hr_des[y].row5, ap_des[y1].row5) == 0)) {
  sum1 = sum1 + atof(hr_des[y].row6);
}
if ((strcmp(hr_des[y].row7, ap_des[y1].row7) == 0) &&
    (strcmp(hr_des[y].row8, ap_des[y1].row8) == 0)) {
  sum1 = sum1 + atof(hr_des[y].row9);
}
```

The desirable factors are matched and the corresponding point is added to sum1
For better user understanding, the name of the HR Manager along with the firm is also stored for display. The total sum of the factors that decides the best match is also calculated.

```
strcpy(hr_suggestions[k].hrname, hr_comp[y].name);
strcpy(hr_suggestions[k].firmname, hr_comp[y].firm);
strcpy(hr_suggestions[k].jobpos, hr_comp[y].row2);
hr_suggestions[k].dessum = sum1;
hr_suggestions[k].totalsum = sum1 + sum;
k++;
```

Rearranging the applicants based on their total sum to display the top 3 best matches.

```
if (flag == 1) {
  for (int x1 = 1; x1 < k; x1++) {
    for (int x2 = 0; x2 < k - x1; x2++) {
      if (hr_suggestions[x2].totalsum >
          hr_suggestions[x2 + 1].totalsum) {
        temp = hr_suggestions[x2];
        hr_suggestions[x2] = hr_suggestions[x2 + 1];
        hr_suggestions[x2 + 1] = temp;
      }
    }
  }
```

Displaying the top 3 candidates

```
printf(
    "\nTop three suitable candidates for the mentioned JOb Vacancy "
    "are as follows:");
for (int x1 = 0; x1 < 3; x1++) {
  printf("\nRank %d.\tName:%s\tEmail:%s\t", x1 + 1,
         hr_suggestions[x1].name, hr_suggestions[x1].email);
}
```

Screenshots of sample code and output:

```c
#include <ctype.h>
#include <stdbool.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <unistd.h>
struct hr_suggestions { … } hr_suggestions[100], temp;

struct ap_suggestions { … } ap_suggestions[100], temp1;

struct hr_comp { … } hr_comp[100];

struct hr_des { … } hr_des[100];

struct ap_comp { … } ap_comp[100];

struct ap_des { … } ap_des[100];
```

```c
FILE *file =
    fopen("hr_details_comp.csv", "a"); // Open th
FILE *file1 =
    fopen("hr_details_des.csv", "a"); // Open the
FILE *file2 =
    fopen("ap_details_comp.csv", "a"); // Open th
FILE *file3 = fopen("ap_details_des.csv", "a");
```

LOGIN:

```c
while (1) {
  if (start == 1) {
    printf("You already have an account!\n");
    char loginID[100], password[100];

    printf("Enter login ID: ");
    scanf("%s", loginID);
    printf("\n");

    printf("Enter password: ");
    scanf("%s", password);
    printf("\n");

    if (checkCredentials(loginID, password, role)) {
      printf("Login successful!\n");
      break;
    } else {
      printf("Invalid login credentials.\n");
      printf("Re-enter details.\n");
    }
  }
```

```
make -s
./main
Welcome to the JHM Hiring Web! We not only open the doors to opportunities but
also to a lifelong career.

To proceed, Enter 1 if you have a login ID.
Enter 2 to create a new account:
1
You already have an account!
Enter login ID: 567890

Enter password: jesus

Invalid login credentials.
Re-enter details.
You already have an account!
Enter login ID: 376554

Enter password: whyami

Login successful!
Welcome HR MANAGER!
```

Comparing the entered details with the details in the file using a user-defined function called checkcredentials. We can also see the contents of the files.

user_details.csv

```
Name,Email,Password,DOB,AP/HR,Mobile,Login _ID
vinayaka,vinayaka@hotmail.com,sammyishere,19/05/02,AP,9876543210,157934
kanan,kanan@symrise.com,whyami,18/04/98,HR,8765432109,376554
prahalad,prahalad@dragaco.com,sheiswell,09/09/99,HR,6543210987,873452
mary,mary@elsevier.com,channamereya,07/07/97,HR,4321098765,324561
Ganesh,Ganesh@qualcom.com,kalachashma,12/02/06,HR,3210987654,432156
bantu,bantu@hotmail.com,womaniya,28/09/00,AP,2109876543,890745
madhmani,madhmani@hotmail.com,mani,29-05-2005,AP,1410065407,221468
joseph,joseph@hotmail.com,great,12-09-2000,HR,1280719729,974096
muruga,muruga@hotmail.com,1234,21-03-2001,AP,1234509876,388650
shoaib,shoaib@google.com,890,03-03-98,HR,1234509876,852864
symrise,shyam@symrise.com,dubai,3-3-03,HR,2000009876,225938
nandi,nandi@hotmail.com,12345,12-3-03,AP,1230984765,946532
suresh,suresh@symrise.com,jithu,12-3-98,HR,1234987655,174710
ghanam,ghanam@htomail.com,abudabi,12-12-4,AP,1234123432,296574
```

login1.csv

```
loginid,Password,AP/HR
157934,sammyishere,AP
376554,whyami,HR
873452,sheiswell,HR
324561,channamereya,HR
432156,kalachashma,HR
890745,womaniya,AP
221468,mani,AP
974096,great,HR
388650,1234,AP
852864,890,HR
225938,dubai,HR
946532,12345,AP
174710,jithu,HR
296574,abudabi,AP
```

HR FUNCTIONALITIES:

AFTER AUTOMATIC DETECTION OF ROLES:

```
if (strcmp(role, "HR\0") == 0) {
  while (1) {
    printf("Welcome HR MANAGER!");
    printf("\nThe following provisions are available for the hiring manager "
           "and his/her team:\n");
    printf("1. To enter a vacancy(with the job description and the "
           "required fields with points assigned)");
    printf("\n2. To edit details of the vacancy entered.");
    printf("\n3. To get top three candidates for entered job along with "
           "accepting or rejecting job profiles suggested.");
    printf(
        "\n4. View details about JOB PROFILES accepted by the HR Manager.");
    printf("\n5. Exit from system");
    printf("\nEnter an option from the above provided features.(1-5):");
    int choice1;
    scanf("%d", &choice1);
```

OPTION 1 - FEW FIELDS OF ENTERING JOB VACANCY

```
// to enter details of the job vacancy
if (choice1 == 1) {

  printf("\nEnter name of Firm, HR Manager:");
  scanf("%s,%s", name1, name2);
  printf("\nEnter login Id:");
  scanf("%d", &login);
  fprintf(file, "%s,%s,%d,", name1, name2, login);
  fprintf(file1, "%s,%s,%d,", name1, name2, login);
```

```
  if (stock_scr == 1) {
    strcpy(field, "Medical Insurance for family\0");
    fprintf(file, "%s,%d,%f", field, insurance, 5 * point);
    fprintf(file1, "%d,%d,%d,", 0, 0, 0);
  } else {
    strcpy(field, "Medical Insurance for family\0");
    fprintf(file1, "%s,%d,%f", field, insurance, 2.5 * point);
    fprintf(file, "%d,%d,%d,", 0, 0, 0);
  }
  fprintf(file, "\n");
  fprintf(file1, "\n");
  printf("\nDetails successfully entered!");
  fclose(file);
  fclose(file1);
}
```

OUTPUT:

```
Login successful!
Welcome HR MANAGER!
The following provisions are available for the hiring manager and his/her team:
1. To enter a vacancy(with the job description and the required fields with points assigned)
2. To edit details of the vacancy entered.
3. To get top three candidates for entered job along with accepting or rejecting job profiles suggested.
4. View details about JOB PROFILES accepted by the HR Manager.
5. Exit from system
Enter an option from the above provided features.(1-5):                      1

Enter name of Firm, HR Manager:symrise,kanan

Enter login Id:376554
```

Entering job vacancy

```
We allow you to change the choice preference and marks allocated for the below mentioned data only to ensure data protecti
on!


Also, to avoid stray users and data protection ,only compulsory data can be changed.Therefore only if any of these fields
have been entered as compulsory field in prior , they can be edited.


1.Job Position
2.Educational Qualification
3.Educational Institution
4.Work Experience
5.Aggregate score
6.Salary range
7.Number of working hours
8.Transportation
9.Shift
10.Work from HomeWARNING:ENTER DATA WITH CAUTION !INCORRECT ENTRY CAN LEAD TO MISMATCH OF DATA!.
```

```
Enter the below details to open a JOB vacancy.
1.Software Developer
2.Doctor(General Physician)
3.Lawyer
4.Teacher
5.Marketing Manager
6.Accountant
7.Buisness Analyst
8.Customer service representative
9.Actuary
10.Web Developer
11.Cashier
12.Data Entry worker
13.Management consultancy
14.Sales Manager
15.Artist
16.Financial Manager
17.Insurance Agent
18.Robotics engineer
19.Hardware engineer
20.Social Media Manager
Choose field of vacancy from the above mentioned options(1-20):10
The Qualifications required for the job are mentioned below:
1. B.E/B.tech
2. B.E/B.tech+ M.s/M.B.A
3. M.B.B.S
4. B.A LLB/JD
5. CFA, FRM
6. B.F.A
Choose qualification required from the above mentioned options(1-6):2

 Is Educational Qualification a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.1
```

```
How many points is alloted this particular field (out of  10):8
The Educational Institution compulsorily required or optional are mentioned below:
1. IIT(Bombay,Madras,Delhi,Kharagpur)
2. NIT(Trichy,Warangal,Suratkal,Roorkee)
3. BITS(Pilani,Goa)
4. Regional Colleges(Top 10)
5. IMU(Kolkata,Chennai,Mumbai)
6. Others.
Choose Educational Institution from the above mentioned options(1-6):3

 Is Educational Institution a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.1


How many points is alloted this particular field (out of  10):6
The Work Experience required for the job are mentioned below:
1. 0-1 years
2. 2-5 years
3. 5-10 years
4. 10+ years
5. No prior work experience required for the job.
Choose  work experience required from the above mentioned options(1-5):4

 Is the Work Experience a compulsorily field   or optional? Enter 1 for compulosry or 0 for optional.1


How many points is alloted this particular field (out of  10):6
```

```
The Aggregate score of 10th and 12th marks required for the job are mentioned below:
1. 95-100 percentage.
2. 85-95 percentage.
3. 70-85 percentage.
4. 60-70 percentage.
5. Less than 70 percentage.
Choose Aggregate score required from the above mentioned options(1-5):4

 Is the Aggregate Score a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.0


How many points is alloted this particular field (out of  10):6

Salary offered per annum inclusive of all the perks are mentioned below:
1. Less than 12 Lakhs
2. 12-25 Lakhs
3. 25-35 Lakhs
4. 35-45 Lakhs
5. More than 45 Lakhs
Choose salary range offered from the above mentioned options(1-5):4

 Is Salary offered a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.0


How many points is alloted this particular field (out of      10):5
```

```
Number of working hours are mentioned below:
1. 8 hours( 5 days a week)
2. 7 hours(5 days a week)
3. 6 hours(7 days a week)
4. 9 hours(6 days a week)
5. More than 9 hours(6 days a week)
Choose Number of working hours from the above mentioned options(1-5):4


How many points is alloted this particular field (out of 10):9

 Is Number of working hours a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.1

Transportation services are mentioned below:
1. Pick up and drop by company bus
2. Only morning Pick up by company pool car
3. Only evening drop by company pool car
4. Only for inter-district office visits
5. No transportation services provided
Choose transportation services offered from the above mentioned options(1-5):5


How many points is alloted this particular field (out of           10):7

 Is Transportation a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.1
```

```
Shift of the job are mentioned below:
1. Day shift
2. Night shift
3. Day shift 3 days a week , night shift 2 days a week
4. Night shift 3 days a week , day shift 2 days a week
5. Flexi Hours( Entry time flexible for the employee)
Choose shift of the job from the above mentioned options(1-5):3


How many points is alloted this particular field (out of           10):7

 Is Shift of the job a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.0

Provision of work from home:
1. Provided
2. Not provided
Choose Provision of work from home from the above mentioned options(1-2):1


How many points is alloted this particular field (out of           10):3

 Is Provision of work from home a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.1
```

```
 Is Provision of work from home a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.1

Employee Type
1.Temporary
2.Permanent
Choose from the above 2 options(1-2):2


How many points is alloted this particular field (out of           10):7

 Is Employee Type a compulsorily field oroptional? Enter 1 for compulsory or 0 for optional1

Job Location of the mentioned vacancy:
1.Chennai
2.Delhi
3.Mumbai
4.Madurai
5.Kolkata
6.Hyderabad
7.Bengaluru
8.Surat
9.Ahmedabad
10.Pune
11.Remote first( NO Geo-location constraints)
Choose Job Location of vacancy from the above mentioned options(1-11):5


How many points is alloted this particular field (out of           10):6
```

```
Number of paid leaves per year:
1. 10 days CL+ 7 days SL
2. 8 days CL+ 5 days SL
3. 5 days CL+ 3 days SL
4. No paid leaves
Choose Number of paid leaves per year from the above mentioned options(1-4):3


How many points is alloted this particular field (out of           10):7

 Is Number of paid leaves per year a compulsorily field or optional? Enter 1 for compulosry or 0 for optional.8

Stock options and incentives
1.Provided
2.Not Provided
Choose from the above 2 options(1-2):1


How many points is alloted this particular field (out of           10):7

 Is this field compulsorily oroptional? Enter 1 for compulsory or 0 for optional0
```

```
Details successfully entered!Welcome HR MANAGER!
The following provisions are available for the hiring manager and his/her team:
1. To enter a vacancy(with the job description and the required fields with points assigned)
2. To edit details of the vacancy entered.
```

VALIDATION OF DETAILS STORED IN THE FILE BEFORE ADDITION:

hr_details_comp.csv

```
score,4,35.000000,Salary
range,3,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0
symrise,shyam,225938,Job Position,18,50.000000,Educational
Qualification,2,20.000000,0,0,0,Work
Experience,4,35.000000,Aggregate score,4,35.000000,Salary
range,3,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,0
```

hr_details_des.csv

```
incentives,1,15.000000,Medical Insurance for
family,2,10.000000
symrise,shyam,225938,0,0,0,0,0,0,Educational
Institution,3,15.000000,0,0,0,0,0,0,0,0,0,Number of working
hours,5,17.500000,Transportation,2,2.500000,0,0,0,Shift,3,17
.500000,Work from Home,3,15.000000,Employee
Type,1,7.500000,Job Location,3,22.500000,Number of paid
leaves,2,17.500000,Stock options and
incentives,2,15.000000,Medical Insurance for
family,2,10.000000
```

AFTER ADDITION:

hr_details_comp.csv

```
range,3,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0
symrise,kanan,376554,Job Position,10,50.000000,Educational
Qualification,1,40.000000,Educational
Institution,1,30.000000,Work
Experience,4,30.000000,0,0,0,0,0,0,Number of working
hours,4,45.000000,Transportation,5,35.000000,0,0,0,Work from
Home,1,15.000000,Employee
Type,2,35.000000,0,0,0,0,0,0,0,0,0,0,0,
```

hr_details_des.csv

```
options and incentives,2,15.000000,Medical Insurance for
family,2,10.000000
8  symrise,kanan,376554,0,0,0,0,0,0,0,0,0,0,0,0,Aggregate
score,4,15.000000,Salary
range,4,12.500000,0,0,0,0,0,0,Shift,3,17.500000,0,0,0,0,0,0,Job
Location,5,15.000000,Number of paid leaves,3,17.500000,Stock
options and incentives,1,17.500000,Medical Insurance for
family,2,17.500000
```

OPTION 2 - TO EDIT THE DETAILS:

```
else if (choice1 == 2) {
  FILE *file = fopen("hr_details_comp.csv", "r");
  struct hr_comp hr_comp[100];

  char line[200];
  int i = 0;

  while (fgets(line, sizeof(line), file)) {…}
  fclose(file);
  printf("\nEnter name of firm:");
  scanf("%s", name10);
  printf("\nEnter name of HR Manager:");
  scanf("%s", hr10);
  printf("\nEnter Job Position number:");
  scanf("%s", job10);
```

TAKING ALL ELEMENTS INTO A STRUCTURE:

Iterating through all the rows and retrieving the row containing the details and changing if the detail is compulsory:

```
// sum for comp sum , sum1 for des sum
int flag = 0, sum = 0, k = 0, sum1 = 0;
for (int y = 0; y < i; y = y + 2) {
  // to find hr details from compulsory file;
  if (strcmp(hr_comp[y].firm, name10) == 0 &&
      strcmp(hr_comp[y].name, hr10) == 0 &&
      strcmp(hr_comp[y].row2, job10) == 0) {
    flag = 1;
    printf("\nWe allow you to change the choice preference and marks "
           "allocated for the below mentioned data only to ensure data "
           "protection!\n");
    printf("\n\nCaution: Modify compulsory fields only for data "
           "protection: incorrect access may lead to data loss.\n\n");
```

Entering the changed data back into the file.
If data is not found , message is displayed

```
        }
        for(int x3=0;x3<i;x3=x3+2){

fprintf(file,"%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s
,%s,%s,% char hr_comp::name[20] %s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s\n",hr_comp[x3].firm,hr_c
omp[x3].name,hr_comp[x3].login,hr_comp[x3].row1,hr_comp[x3].row2,hr_comp[x3].row3,hr_comp[x
3].row4,hr_comp[x3].row5,hr_comp[x3].row6,hr_comp[x3].row7,hr_comp[x3].row8,hr_comp[x3].row
9,hr_comp[x3].row10,hr_comp[x3].row11,hr_comp[x3].row12,hr_comp[x3].row13,hr_comp[x3].row14
,hr_comp[x3].row15,hr_comp[x3].row16,hr_comp[x3].row17,hr_comp[x3].row18,hr_comp[x3].row19,
hr_comp[x3].row20,hr_comp[x3].row21,hr_comp[x3].row22,hr_comp[x3].row23,hr_comp[x3].row24,h
r_comp[x3].row25,hr_comp[x3].row26,hr_comp[x3].row27,hr_comp[x3].row28,hr_comp[x3].row29,hr
_comp[x3].row30,hr_comp[x3].row31,hr_comp[x3].row32,hr_comp[x3].row33,hr_comp[x3].row34,hr_
comp[x3].row35,hr_comp[x3].row36,hr_comp[x3].row37,hr_comp[x3].row38,hr_comp[x3].row39,hr_c
omp[x3].row40,hr_comp[x3].row41,hr_comp[x3].row42,hr_comp[x3].row43,hr_comp[x3].row44,hr_co
mp[x3].row45);
```

Condition to find end of file and stop entry of values:

```
counter++;
if (counter == ((i / 2))) {
  fclose(file);
  printf("\n\nUpdation occured succesfully!\n\n");
  printf("\n\n----------------------------------------\n\n");
  break;
}
```

OPTION 3 - PERFORMING MATCHING ALGORITHM AND ACCEPTING/REJECTING AN OFFER

Retrieving elements from the structure based on entered details such as name, firm and job position and entering the matched elements into the structure

```
for (int y = 0; y < i; y = y + 2) {
  // to find hr details from compulsory file;
  if (strcmp(hr_comp[y].firm, name10) == 0 &&
      strcmp(hr_comp[y].name, hr10) == 0 &&
      strcmp(hr_comp[y].row2, job10) == 0) {
    flag = 1;
    // matching details of the compulsory file
    for (int y1 = 0; y1 < i; y1++) {
      sum = 0;
      sum1 = 0;
      if ((strcmp(hr_comp[y].row4, ap_comp[y1].row4) == 0) &&
          (strcmp(hr_comp[y].row1, ap_comp[y1].row1) == 0) &&
          (strcmp(hr_comp[y].row2, ap_comp[y1].row2) == 0) &&
          (strcmp(hr_comp[y].row5, ap_comp[y1].row5) == 0) &&
          (strcmp(hr_comp[y].row7, ap_comp[y1].row7) == 0) &&
          (strcmp(hr_comp[y].row8, ap_comp[y1].row8) == 0) &&
          (strcmp(hr_comp[y].row10, ap_comp[y1].row10) == 0) &&
          (strcmp(hr_comp[y].row11, ap_comp[y1].row11) == 0) &&
          (strcmp(hr_comp[y].row13, ap_comp[y1].row13) == 0) &&
          (strcmp(hr_comp[y].row14, ap_comp[y1].row14) == 0) &&
          (strcmp(hr_comp[y].row16, ap_comp[y1].row16) == 0) &&
          (strcmp(hr_comp[y].row17, ap_comp[y1].row17) == 0) &&
          (strcmp(hr_comp[y].row19, ap_comp[y1].row19) == 0) &&
```

This condition checking happens for all the rows.Once all the compulsory factors have matched,

Fields like name of APPLICANT , email of APPLICANT and their compulsory matched sum is stored.

```
hr_suggestions[k].compsum = sum;
strcpy(hr_suggestions[k].name, ap_comp[y1].name);
char email[30];
strcpy(email, ap_comp[y1].name);
strcat(email, "@hotmail.com");
strcpy(hr_suggestions[k].email, email);
if ((strcmp(hr_des[y].row1, ap_des[y1].row1) == 0) &&
    (strcmp(hr_des[y].row2, ap_des[y1].row2) == 0)) {
  sum1 = sum1 + atof(hr_des[y].row3);
}
if ((strcmp(hr_des[y].row4, ap_des[y1].row4) == 0) &&
    (strcmp(hr_des[y].row5, ap_des[y1].row5) == 0)) {
  sum1 = sum1 + atof(hr_des[y].row6);
}
if ((strcmp(hr_des[y].row7, ap_des[y1].row7) == 0) &&
    (strcmp(hr_des[y].row8, ap_des[y1].row8) == 0)) {
  sum1 = sum1 + atof(hr_des[y].row9);
}
```

The desirable factors are matched and the corresponding point is added to sum1
For better user understanding, the name of the HR Manager along with the firm is also stored for display. The total sum of the factors that decides the best match is also calculated.

```
strcpy(hr_suggestions[k].hrname, hr_comp[y].name);
strcpy(hr_suggestions[k].firmname, hr_comp[y].firm);
strcpy(hr_suggestions[k].jobpos, hr_comp[y].row2);
hr_suggestions[k].dessum = sum1;
hr_suggestions[k].totalsum = sum1 + sum;
k++;
```

Rearranging the applicants based on their total sum to display the top 3 best matches.

```
if (flag == 1) {
  for (int x1 = 1; x1 < k; x1++) {
    for (int x2 = 0; x2 < k - x1; x2++) {
      if (hr_suggestions[x2].totalsum >
          hr_suggestions[x2 + 1].totalsum) {
        temp = hr_suggestions[x2];
        hr_suggestions[x2] = hr_suggestions[x2 + 1];
        hr_suggestions[x2 + 1] = temp;
      }
    }
  }
```

Displaying the top 3 candidates

```
printf(
    "\nTop three suitable candidates for the mentioned JOb Vacancy "
    "are as follows:");
for (int x1 = 0; x1 < 3; x1++) {
  printf("\nRank %d.\tName:%s\tEmail:%s\t", x1 + 1,
         hr_suggestions[x1].name, hr_suggestions[x1].email);
}
```

Accepting or Rejecting the preference and the preference is stored to enable future reference for the HR Manager

```
FILE *file4 = fopen("hr_ranks.csv", "a");
fprintf(file4, "%s,%s,%s", name10, hr10, job10);
printf("\nEnter number of profiles you would like to accept:");
int accept;
scanf("%d", &accept);
for (int y4 = 0; y4 < accept; y4++) {
  printf(
      "\nEnter Rank number of the profile you would like to continue "
      "and initiate recruitment process with(Rank number 1,2..):");
  int rank;
  scanf("%d", &rank);
  if (rank == 1) {
    fprintf(file4, ",%s,%s", hr_suggestions[0].name,
            hr_suggestions[0].email);
  }
  if (rank == 2) {
    fprintf(file4, ",%s,%s", hr_suggestions[1].name,
            hr_suggestions[1].email);
  }

  if (rank == 3) {
    fprintf(file4, ",%s,%s", hr_suggestions[2].name,
            hr_suggestions[2].email);
```

OUTPUT:

```
ted.
4. View details about JOB PROFILES accepted by the HR Manager.
5. Exit from system
Enter an option from the above provided features.(1-5):3

Enter name of firm:symrise

Enter name of HR Manager:kanan

Enter Job Position number:1

Top three suitable candidates for the mentioned JOb Vacancy are as follows:
Rank 1. Name:vinayaka    Email:vinayaka@hotmail.com
Rank 2. Name:muruga Email:muruga@hotmail.com
Rank 3. Name:nandi  Email:nandi@hotmail.com
Enter number of profiles you would like to accept:2

Enter Rank number of the profile you would like to continue and initiate recruitment process with(Ra
nk number 1,2..):1

Enter Rank number of the profile you would like to continue and initiate recruitment process with(Ra
nk number 1,2..):2

Preferences noted.
Welcome HR MANAGER!
The following provisions are available for the hiring manager and his/her team:
```

VERIFYING WITH THE DETAILS ABOUT THE JOB VACANCY AND DISPLAYED OPTIONS:

Contents of the file

hr_details_des.csv

```
1  symrise,kanan,376554,0,0,0,0,0,0,Educational
   Institution,1,15.000000,0,0,0,0,0,0,0,0,Number of working
   hours,1,17.500000,Transportation,2,2.500000,Shift,3,17.50000
   0,Work from Home,3,15.000000,Employee Type,1,7.500000,Job
   Location,2,22.500000,Number of paid
   leaves,1,17.500000,Stock options and
   incentives,1,15.000000,Medical Insurance for
   family,1,10.000000
2  dragaco,prahalad,873452,0,0,0,0,0,0,Educational
```

hr_details_comp.csv

```
1  symrise,kanan,376554,Job Position,1,50.000000,Educational
   Qualification,1,20.000000,0,0,0,Work
   Experience,4,70,Aggregate score,2,35.000000,Salary
   range,4,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0
2  dragaco,prahalad,873452,Job
```

APPLICANT DETAILS:

ap_details_comp.csv

```
1  0,muruga,388650,Job Position,1,0,Educational
   Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
   score,2,0,Salary
   range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
2  0,vinayaka,157934,Job Position,1,0,Educational
   Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
   score,2,0,Salary
   range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3  0,nandi,946532,Job Position,1,0,Educational
   Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
   score,2,0,Salary
   range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
4  0,ghanam,296574,Job Position,1,0,Educational
   Qualification,3,0,0,0,0,Work Experience,3,0,Aggregate
   score,2,0,Salary
   range,4,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
5  0,Deepika,903808,Job Position,19,0,Educational
   Qualification,1,0,0,0,0,Work Experience,2,0,Aggregate
   score,4,0,Salary
   range,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,Medical
   Insurance for family,1,0
6
```

ap_details_des.csv

```
1  0,muruga,388650,0,0,0,0,0,0,Educational Institution,2,0,0,0,0,0,0,0,0,0,0,Number
   of working hours,1,0,Transportation,3,0,0,0,0,Shift,3,0,Work from
   Home,3,0,Employee Type,1,0,Job Location,2,0,Number of paid leaves,1,0,Stock
   options and incentives,1,0,Medical Insurance for family,1,0
2  0,vinayaka,157934,0,0,0,0,0,0,Educational
   Institution,1,0,0,0,0,0,0,0,0,0,0,Number of working
   hours,1,0,Transportation,2,0,0,0,0,Shift,3,0,Work from Home,3,0,Employee
   Type,1,0,Job Location,2,0,Number of paid leaves,1,0,Stock options and
   incentives,2,0,Medical Insurance for family,1,0
3  0,nandi,946532,0,0,0,0,0,0,Educational Institution,1,0,0,0,0,0,0,0,0,0,0,Number
   of working hours,2,0,Transportation,1,0,0,0,0,Shift,3,0,Work from
   Home,3,0,Employee Type,1,0,Job Location,2,0,Number of paid leaves,3,0,Stock
   options and incentives,2,0,Medical Insurance for family,1,0
4  0,ghanam,296574,0,0,0,0,0,0,Educational Institution,3,0,0,0,0,0,0,0,0,0,0,Number
   of working hours,2,0,Transportation,3,0,0,0,0,Work from
   Home,3,17.500000,Shift,3,0,Employee Type,1,0,Job Location,2,0,Number of paid
   leaves,3,0,Stock options and incentives,2,0,Medical Insurance for family,1,0
5  0,Deepika,903808,0,0,0,Educational Institution,1,0,0,0,0,0,0,0,0,0,0,Number of
   working hours,3,0,Transportation,5,0,Shift,1,0,Work from home,2,0,Employee
   Type,1,0,Job Location,9,0,Number of paid leaves,2,0,Stock options and
   incentives,2,0,0,0,0,0,0,0
6
```

OPTION 4 -When choice 4 is entered to view the previously chosen preference:

```
while (fgets(line, sizeof(line), file4)) {
  // Create a copy of the line for tokenization
  char line_copy[256];
  strcpy(line_copy, line);

  // Split the line into tokens
  token = strtok(line_copy, ",");

  // Check if the first three values match the desired values
  if (token != NULL && strcmp(token, name11) == 0) {
    token = strtok(NULL, ",");
    if (token != NULL && strcmp(token, hr11) == 0) {
      token = strtok(NULL, ",");
      if (token != NULL && strcmp(token, job11) == 0) {
        flag3 = 1;
        printf("\nDetails about the HR manager along with the JOB "
               "position entered and profiles accpeted \n");

        // Print all details in the found row
        token = strtok(line, ",");
        while (token != NULL) { … }
        break; // Exit the loop after finding the row
      }
```

OUTPUT:

```
 Job profiles suggested.
4. View details about JOB PROFILES accepted by the HR Manager.
5. Exit from system
Enter an option from the above provided features.(1-5):4

Enter name of firm:symrise

Enter name of HR Manager:kanan

Enter Job Position number:1

Details about the HR manager along with the JOB position entered and profiles ac
cpeted
symrise
kanan
1
vinayaka
vinayaka@hotmail.com
muruga
muruga@hotmail.com

Welcome HR MANAGER!
The following provisions are available for the hiring manager and his/her team:
```

OPTION 5- Exiting from the system

```
3. To get top three candidates for entered job along with accepting or rejecting job profiles sugges
ted.
4. View details about JOB PROFILES accepted by the HR Manager.
5. Exit from system
Enter an option from the above provided features.(1-5):5


Thank you so much for visiting our portal. We hope that our portal opened new doors to exciting oppo
rtunities.Good luck:)

exit status 1
```

SOME FIELDS OF CREATING AN ACCOUNT

```c
else if (start == 2) {
  char name[4], email[50], password[20], cpassword[20], dob[10];
  int mobile;
  printf("Enter name of user:");
  scanf("%s", name);
  printf("\n");
  printf("Enter email-id of user:");
  scanf("%s", email);
  printf("\n");
  printf("Enter password to be set:");
  scanf("%s", password);
  printf("\n");
  printf("Confirm password:");
  scanf("%s", cpassword);
  printf("\n");
```

APPLICANT

```
> make -s
> ./main
Welcome to the JHM Hiring Web! We not only open the doors to
 opportunities but also to a lifelong career.

To proceed, Enter 1 if you have a login ID.
Enter 2 to create a new account:
2
Enter name of user:Deepika

Enter email-id of user:deepika@hotmail.com

Enter password to be set:god

Confirm password:god1

Re-enter detail.
Confirm password:god

Enter Date of Birth of user:21-03-2000

Enter AP for applicant(or) HR for hiring manager:AP

Enter mobile number: 9876512340

Generated Login ID: 903808
Login ID generated and stored successfully!


Welcome APPLICANT!

--------------------------------------
```

AFTER ADDITION OF ACCOUNT:

user_details.csv

```
98,HR,1234987655,174710
15 ghanam,ghanam@htomail.com,abudabi,12-12-
4,AP,1234123432,296574
16 Deepika,deepika@hotmail.com,god,21-03-
2000,AP,1286577748,903808
17
```

login1.csv

```
14 174710,jithu,HR
15 296574,abudabi,AP
16 903808,god,AP
```

CONTAINS OF THE DETAILS ABOUT APPLICANT;

ap_details_comp.csv

```
1  0,muruga,388650,Job Position,1,0,Educational
   Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
   score,2,0,Salary
   range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0
2  0,vinayaka,157934,Job Position,1,0,Educational
   Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
   score,2,0,Salary
   range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0
3  0,nandi,946532,Job Position,1,0,Educational
   Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
   score,2,0,Salary
   range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0
4  0,ghanam,296574,Job Position,1,0,Educational
   Qualification,3,0,0,0,0,Work Experience,3,0,Aggregate
   score,2,0,Salary
   range,4,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
   ,0,0
```

ap_details_des.csv

```
1  0,muruga,388650,0,0,0,0,0,0,Educational
   Institution,2,0,0,0,0,0,0,0,0,0,0,Number of working
   hours,1,0,Transportation,3,0,0,0,0,Shift,3,0,Work from Home,3,0,Employee
   Type,1,0,Job Location,2,0,Number of paid leaves,1,0,Stock options and
   incentives,1,0,Medical Insurance for family,1,0
2  0,vinayaka,157934,0,0,0,0,0,0,Educational
   Institution,1,0,0,0,0,0,0,0,0,0,0,Number of working
   hours,1,0,Transportation,2,0,0,0,0,Shift,3,0,Work from Home,3,0,Employee
   Type,1,0,Job Location,2,0,Number of paid leaves,1,0,Stock options and
   incentives,2,0,Medical Insurance for family,1,0
3  0,nandi,946532,0,0,0,0,0,0,Educational
   Institution,1,0,0,0,0,0,0,0,0,0,0,Number of working
   hours,2,0,Transportation,1,0,0,0,0,Shift,3,0,Work from Home,3,0,Employee
   Type,1,0,Job Location,2,0,Number of paid leaves,3,0,Stock options and
   incentives,2,0,Medical Insurance for family,1,0
4  0,ghanam,296574,0,0,0,0,0,0,Educational
   Institution,3,0,0,0,0,0,0,0,0,0,0,Number of working
   hours,2,0,Transportation,3,0,0,0,0,Work from
   Home,3,17.500000,Shift,3,0,Employee Type,1,0,Job Location,2,0,Number of
   paid leaves,3,0,Stock options and incentives,2,0,Medical Insurance for
   family,1,0
```

CHOICES AVAILABLE FOR APPLICANT:

```
else if (strcmp(role, "AP\0") == 0) {
  while (1) {
    printf("\n\nWelcome APPLICANT!");
    printf("\n\n----------------------------------------\n\n");
    printf("\nThe following provisions are available for the applicant:\n");
    printf("\n1. To enter JOB profile");
    printf("\n2. To edit details of the JOB PROFILE entered.");
    printf("\n3. To get top three job vacancies for entered job profile "
           "along with "
           "accepting or rejecting job vacancies suggested.");
    printf("\n4.View details about JOB PROFILES accepted by the HR Manager.");
    printf("\n5.Exit from the system");
    printf("\nEnter an option from the above provided features.(1-5):");
    printf("\n\n----------------------------------------\n\n");

    int choice2;
    scanf("%d", &choice2);
    // to enter details of the job profile
    if (choice2 == 1) { … }

    else if (choice2 == 2) { … }
```

OPTION 3- TRY TO COMPARE DETAILS WITHOUT ENTERING JOB PROFILE

```
----------------------------------------


The following provisions are available for the applicant:

1. To enter JOB profile
2. To edit details of the JOB PROFILE entered.
3. To get top three job vacancies for entered job profile along wi
th accepting or rejecting job vacancies suggested.
4.View details about JOB PROFILES accepted by the HR Manager.
5.Exit from the system
Enter an option from the above provided features.(1-5):

----------------------------------------

2

Enter name of Applicant:Deepika

Enter Job Position number:4

Sorry ,above mentioned details along with Job vacancy not found.


Welcome APPLICANT!

----------------------------------------
```

OPTION 1 - ENTERING JOB PROFILE:

```
Enter the below details to open a JOB Profile.
1.Software Developer
2.Doctor(General Physician)
3.Lawyer
4.Teacher
5.Marketing Manager
6.Accountant
7.Business Analyst
8.Customer service representative
9.Actuary
10.Web Developer
11.Cashier
12.Data Entry worker
13.Management consultancy
14.Sales Manager
15.Artist
16.Financial Manager
17.Insurance Agent
18.Robotics engineer
19.Hardware engineer
20.Social Media Manager
Choose career field from the above mentioned options(1-20):19
```

```
The Work Experience of the applicant are mentioned below:
1. 0-1 years
2. 2-5 years
3. 5-10 years
4. 10+ years
5. No prior work experience required for the job.
Choose  work experience from the above mentioned options(1-5):3

 Is the this field compulsorily or optional? Enter 1 for compulosry or 0 for
 optional.1
The Aggregate score of 10th and 12th marks are mentioned below:
1. 95-100 percentage.
2. 85-95 percentage.
3. 70-85 percentage.
4. 60-70 percentage.
5. Less than 70 percentage.
Choose Aggregate score from the above mentioned options(1-5):4

 Is this field compulsorily or optional? Enter 1 for compulosry or 0 for opt
ional.1
```

```
Stock options and incentives
1.Required
2.Not Required
Enter your choice(1-2):2

 Is this field compulsorily oroptional? Enter 1 for compulsory or 0 for opti
onal0

Medical Insurance for family
1.Required
2.Not Required
Enter your choice(1-2):1

 Is this field compulsory oroptional? Enter 1 for compulsory or 0 for option
al1

Entered details and preferences successfully


Welcome APPLICANT!
```

Addition reflecting in the file

```
ap_details_comp.csv
score,2,0,Salary
range,4,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,Deepika,903808,Job Position,19,0,Educational
Qualification,1,0,0,0,0,Work Experience,3,0,Aggregate
score,4,0,Salary
range,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,Medical
Insurance for family,1,0
```

```
ap_details_des.csv
Location,2,0,Number of paid leaves,3,0,Stock options and
incentives,2,0,Medical Insurance for family,1,0
0,Deepika,903808,0,0,0,Educational
Institution,1,0,0,0,0,0,0,0,0,0,0,Number of working
hours,3,0,Transportation,5,0,Shift,1,0,Work from
Home,2,0,Employee Type,1,0,Job Location,9,0,Number of paid
leaves,2,0,Stock options and incentives,2,0,0,0,0,0,0,0,0
```

OPTION 2 - EDITING THE ENTER JOB

```
else if (choice2 == 2) {
  FILE *file2 = fopen("ap_details_comp.csv", "r");

  struct ap_comp ap_comp[100];

  char line[200];
  int i = 0;
  while (fgets(line, sizeof(line), file2)) {
    char *field = strtok(line, ",");
    int field_count = 0;
    while (field) {…}
    i++;
  }
  fclose(file2);
  char ap10[30], apjob10[30];
  printf("\nEnter name of Applicant:");
  scanf("%s", ap10);
  printf("\nEnter Job Position number:");
  scanf("%s", apjob10);
```

```
for (int y1 = 0; y1 < i; y1++) {
  // to find hr details from compulsory file;
  if ((strcmp(ap_comp[y1].name, ap10) == 0) &&
      (strcmp(ap_comp[y1].row2, apjob10) == 0)) {

    flag = 1;
    printf("\nWe allow you to change the choice preference
rks "
           "allocated for the below mentioned data only to
. data "
```

```
if (choice_field == 8) {
  printf("\n1. Pick up and drop by company bus");
  printf("\n2. Only morning Pick up by company pool

  printf("\n3. Only evening drop by company pool car");
  printf("\n4. Only for inter-district office visits");
  printf("\n5. No transportation services provided");
  printf("\nEnter new preference number:\n");
  char pref[2];
  scanf("%s", pref);
  strcpy(ap_comp[y1].row23, pref);
  printf("\n\n---------------------------------------

}
```

```
      if (flag == 1) {
        int counter1 = 0;
        file2 = fopen("ap_details_comp.csv", "w");
        for (int x4 = 0; x4 < i; x4++) {
          fprintf(
              file2,

"%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,%s,"
```

```
                ap_comp[x4].firm, ap_comp[x4].name,
ap_comp[x4].login,
                ap_comp[x4].row1, ap_comp[x4].row2,
ap_comp[x4].row3,
                ap_comp[x4].row4, ap_comp[x4].row5,
ap_comp[x4].row6,
                ap_comp[x4].row7, ap_comp[x4].row8,
ap_comp[x4].row9,
                ap_comp[x4].row10, ap_comp[x4].row11,
ap_comp[x4].row12,
                ap_comp[x4].row13, ap_comp[x4].row14,
ap_comp[x4].row15,
                ap_comp[x4].row16, ap_comp[x4].row17,
ap_comp[x4].row18,
                ap_comp[x4].row19, ap_comp[x4].row20,
```

```
  if (counter1 == i - 1) {
    fclose(file2);
    printf("\n\nUpdation occured succesfully!\n\n");
    printf("\n\n---------------------------------------

    break;
  }
```

OUTPUT:

```
Welcome APPLICANT!

--------------------------------------


The following provisions are available for the applicant:

1. To enter JOB profile
2. To edit details of the JOB PROFILE entered.
3. To get top three job vacancies for entered job profile along with
 accepting or rejecting job vacancies suggested.
4.View details about JOB PROFILES accepted by the HR Manager.
5.Exit from the system
Enter an option from the above provided features.(1-5):

--------------------------------------

2

Enter name of Applicant:Deepika

Enter Job Position number:19

We allow you to change the choice preference and marks allocated for
 the below mentioned data only to ensure data protection!
```

```
Which of the following fields do you want to change the value of?
1.Job Position
2.Educational Qualification
3.Educational Institution
4.Work Experience
5.Aggregate score
6.Salary range
7.Number of working hours
8.Transportation
9.Shift
10.Work from Home
Choose a field number(1-10):4


--------------------------------------


1. 0-1 years
2. 2-5 years
3. 5-10 years
4. 10+ years
5. No prior work experience required for the job.Enter new preferenc
e number:
2
```

```
Updation occured succesfully!


----------------------------------------


Welcome APPLICANT!

--------------------------------------
```

Change reflected in the file:

```
ap_details_comp.csv
    range,4,30,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    ,0,0,0,0,0
5   0,Deepika,903808,Job Position,19,0,Educational
    Qualification,1,0,0,0,0,Work Experience,2,0,Aggregate
    score,4,0,Salary
    range,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
    0,0,Medical Insurance for family,1,0
6
```

After adding profile, vacancy is searched:

```
5.Exit from the system
Enter an option from the above provided features.(1-5):

---------------------------------------

3

Enter name of applicant:Deepika

Enter Job Position number:19

Found applicant details!.
No suitable job vacancy is currently available.


Welcome APPLICANT!
```

New user

OPTION 3 - MATCHING ALGORITHM (APPLICANT)

```
Enter 2 to create a new account:
1
You already have an account!
Enter login ID: 157934

Enter password: sammyishere

Login successful!


Welcome APPLICANT!

---------------------------------------


The following provisions are available for the applicant
:

1. To enter JOB profile
2. To edit details of the JOB PROFILE entered.
3. To get top three job vacancies for entered job profil
e along with accepting or rejecting job vacancies sugges
ted.
4.View details about JOB PROFILES accepted by the HR Man
ager.
5.Exit from the system
Enter an option from the above provided features.(1-5):

---------------------------------------

3

Enter name of applicant:vinayaka
```

To enter the values from the files to the structure

```
while (fgets(line, sizeof(line), file)) {
  char *field = strtok(line, ",");
  int field_count = 0;
  while (field) {
    if (field_count == 0) {
      strncpy(hr_comp[i].firm, field, sizeof(hr_comp[i].firm) - 1);
      hr_comp[i].firm[sizeof(hr_comp[i].firm) - 1] = '\0';
    }
    if (field_count == 1) {
      strncpy(hr_comp[i].name, field, sizeof(hr_comp[i].name) - 1);
      hr_comp[i].name[sizeof(hr_comp[i].name) - 1] = '\0';
    }
    if (field_count == 2) {
      strncpy(hr_comp[i].login, field, sizeof(hr_comp[i].login) - 1);
      hr_comp[i].login[sizeof(hr_comp[i].login) - 1] = '\0';
    }
    if (field_count == 3) {
      strncpy(hr_comp[i].row1, field, sizeof(hr_comp[i].row1) - 1);
      hr_comp[i].row1[sizeof(hr_comp[i].row1) - 1] = '\0';
    }
```

The above code is similarly implemented for all the files needed for matching.

```
while (fgets(line, sizeof(line), file)) { … }
i = 0;
while (fgets(line, sizeof(line), file1)) { … }
i = 0;
while (fgets(line, sizeof(line), file2)) { … }
i = 0;
while (fgets(line, sizeof(line), file3)) { … }
```

To get the best suitable vacancies , the particular row is first retrieved using the below logic:

```
char ap10[30], apjob10[30];
printf("\nEnter name of applicant:");
scanf("%s", ap10);
printf("\nEnter Job Position number:");
scanf("%s", apjob10);
// sum for comp sum , sum1 for des sum
int flag = 0, sum = 0, k = 0, sum1 = 0, flag2 = 0;
for (int y = 0; y < i / 2; y++) {
  // to find applicant details from compulsory file;
  if ((strcmp(ap_comp[y].name, ap10) == 0) &&
      (strcmp(ap_comp[y].row2, apjob10) == 0)) {
    printf("\nFound applicant details!.");
    flag = 1;
```

As implemented in prior case of HR , all the fields are compared to ensure matching of compulsory fields , followed by finding the sums of the common desirable fields if and only if all the compulsory fields match

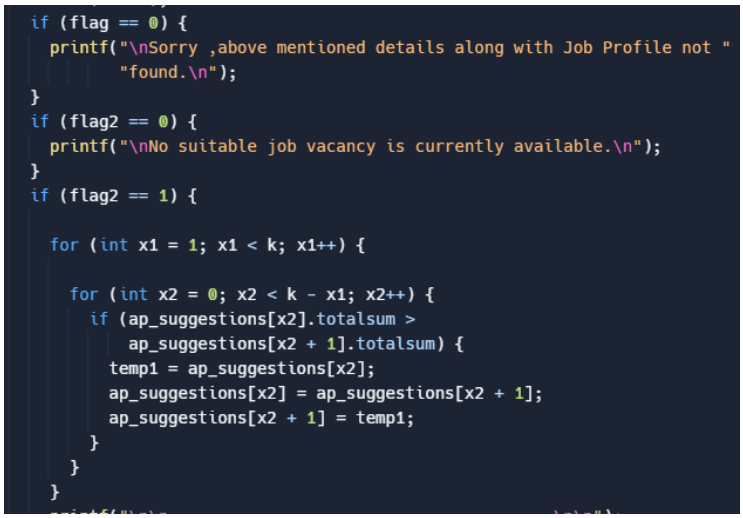
```
// matching details of the compulsory file
for (int y1 = 0; y1 < i; y1 = y1 + 2) {
  sum = 0;
  sum1 = 0;
  if ((strcmp(ap_comp[y].row4, hr_comp[y1].row4) == 0) &&
      (strcmp(ap_comp[y].row1, hr_comp[y1].row1) == 0) &&
      (strcmp(ap_comp[y].row2, hr_comp[y1].row2) == 0) &&
      (strcmp(ap_comp[y].row5, hr_comp[y1].row5) == 0) &&
      (strcmp(ap_comp[y].row7, hr_comp[y1].row7) == 0) &&
      (strcmp(ap_comp[y].row8, hr_comp[y1].row8) == 0) &&
      (strcmp(ap_comp[y].row10, hr_comp[y1].row10) == 0) &&
      (strcmp(ap_comp[y].row11, hr_comp[y1].row11) == 0) &&
      (strcmp(ap_comp[y].row13, hr_comp[y1].row13) == 0) &&
      (strcmp(ap_comp[y].row14, hr_comp[y1].row14) == 0) &&
      (strcmp(ap_comp[y].row16, hr_comp[y1].row16) == 0) &&
      (strcmp(ap_comp[y].row17, hr_comp[y1].row17) == 0) &&
      (strcmp(ap_comp[y].row19, hr_comp[y1].row19) == 0) &&
      (strcmp(ap_comp[y].row20, hr_comp[y1].row20) == 0) &&
      (strcmp(ap_comp[y].row22, hr_comp[y1].row22) == 0) &&
      (strcmp(ap_comp[y].row23, hr_comp[y1].row23) == 0) &&
```

```
          atof(hr_comp[y1].row45) + atof(hr_comp[y1].row45);
      ap_suggestions[k].compsum = sum;
      strcpy(ap_suggestions[k].name, hr_comp[y1].name);
      strcpy(ap_suggestions[k].firm, hr_comp[y1].firm);
      char email[30];
      strcpy(email, hr_comp[y1].name);
      strcat(email, "@");
      strcat(email, hr_comp[y1].firm);
      strcat(email, ".com");
      strcpy(ap_suggestions[k].email, email);
      if ((strcmp(ap_des[y].row1, hr_des[y1].row1) == 0) &&
          (strcmp(ap_des[y].row2, hr_des[y1].row2) == 0)) {
        sum1 = sum1 + atof(hr_des[y1].row3);
      }
      if ((strcmp(ap_des[y].row4, hr_des[y1].row4) == 0) &&
          (strcmp(ap_des[y].row5, hr_des[y1].row5) == 0)) {
        sum1 = sum1 + atof(hr_des[y1].row6);
      }
      if ((strcmp(ap_des[y].row7, hr_des[y1].row7) == 0) &&
          (strcmp(ap_des[y].row8, hr_des[y1].row8) == 0)) {
        sum1 = sum1 + atof(hr_des[y1].row9);
      }
```

The total sum of points calculated from both compulsory and matching fields are stored.

```
    sum1 = sum1 + atof(hr_des[y1].row30);
  }
  ap_suggestions[k].dessum = sum1;
  ap_suggestions[k].totalsum = sum1 + sum;
  k++;
}
```

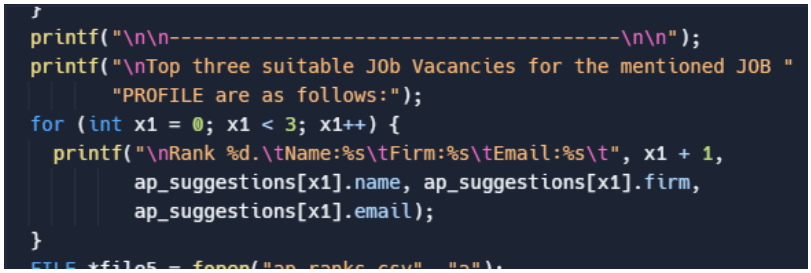
The matched details are rearranged to display only the top three matches



```
if (flag == 0) {
  printf("\nSorry ,above mentioned details along with Job Profile not "
         "found.\n");
}
if (flag2 == 0) {
  printf("\nNo suitable job vacancy is currently available.\n");
}
if (flag2 == 1) {

  for (int x1 = 1; x1 < k; x1++) {

    for (int x2 = 0; x2 < k - x1; x2++) {
      if (ap_suggestions[x2].totalsum >
          ap_suggestions[x2 + 1].totalsum) {
        temp1 = ap_suggestions[x2];
        ap_suggestions[x2] = ap_suggestions[x2 + 1];
        ap_suggestions[x2 + 1] = temp1;
      }
    }
  }
```

Also,If the applicant's profile does not match with any vacancy , a suitable message is displayed.

Details of the top three matches are displayed like Name of Firm , HR Manager , email id and Rank



```
printf("\n\n---------------------------------------\n\n");
printf("\nTop three suitable JOb Vacancies for the mentioned JOB "
       "PROFILE are as follows:");
for (int x1 = 0; x1 < 3; x1++) {
  printf("\nRank %d.\tName:%s\tFirm:%s\tEmail:%s\t", x1 + 1,
         ap_suggestions[x1].name, ap_suggestions[x1].firm,
         ap_suggestions[x1].email);
}
```

The given options can also be accepted or rejected by the applicant and this preference is safely stored for further reference.

```c
FILE *file5 = fopen("ap_ranks.csv", "a");
fprintf(file5, "%s,%s", ap10, apjob10);
printf("\nEnter number of vacancies you would like to apply:");
int accept;
scanf("%d", &accept);
for (int y4 = 0; y4 < accept; y4++) {
  printf(
      "\nEnter Rank number of the vacancy you would like to continue "
      "and initiate application process with(Rank number 1,2..):");
  int rank;
  scanf("%d", &rank);
  if (rank == 1) {
    fprintf(file5, ",%s,%s,%s", ap_suggestions[0].firm,
            ap_suggestions[0].name, ap_suggestions[0].email);
  }
  if (rank == 2) {
    fprintf(file5, ",%s,%s,%s", ap_suggestions[1].firm,
            ap_suggestions[1].name, ap_suggestions[1].email);
  }

  if (rank == 3) {
    fprintf(file5, ",%s,%s,%s", ap_suggestions[2].firm,
```

OUTPUT of the matching algorithm:

```
----------------------------------------

3

Enter name of applicant:vinayaka

Enter Job Position number:1

Found applicant details!.

----------------------------------------


Top three suitable JOb Vacancies for the mentioned JOB PROFILE are as follows:
Rank 1. Name:kanan  Firm:symrise    Email:kanan@symrise.com
Rank 2. Name:prahalad   Firm:dragaco    Email:prahalad@dragaco.com
Rank 3. Name:mary   Firm:elsevier   Email:mary@elsevier.com
Enter number of vacancies you would like to apply:3

Enter Rank number of the vacancy you would like to continue and initiate application p
rocess with(Rank number 1,2..):1

Enter Rank number of the vacancy you would like to continue and initiate application p
rocess with(Rank number 1,2..):3

Enter Rank number of the vacancy you would like to continue and initiate application p
rocess with(Rank number 1,2..):2

Preferences noted.
```

Verifying with the content of the files:

```
ap_details_comp.csv
2   0,vinayaka,157934,Job Position,1,0,Educational
    Qualification,1,0,0,0,0,Work Experience,4,0,Aggregate
    score,2,0,Salary
    range,4,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
3   0,nandi,946532,Job Position,1,0,Educational
```

ap_details_des.csv

```
for family,1,0
0,vinayaka,157934,0,0,0,0,0,0,Educational
Institution,1,0,0,0,0,0,0,0,0,0,0,Number of working
hours,1,0,Transportation,2,0,0,0,0,Shift,3,0,Work from
Home,3,0,Employee Type,1,0,Job Location,2,0,Number of paid
leaves,1,0,Stock options and incentives,2,0,Medical Insurance
for family,1,0
```

JOB VACANCY details to verify the matching fields

hr_details_comp.csv

```
symrise,kanan,376554,Job Position,1,50.000000,Educational
Qualification,1,20.000000,0,0,0,Work Experience,4,35,Aggregate
score,2,35.000000,Salary
range,4,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0
dragaco,prahalad,873452,Job Position,1,50.000000,Educational
Qualification,1,20.000000,0,0,0,Work Experience,4,30,Aggregate
score,2,35.000000,Salary
range,4,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0
qualcom,Ganesh,432156,Job Position,1,50.000000,Educational
Qualification,1,10.000000,0,0,0,Work Experience,4,25,Aggregate
score,2,35.000000,Salary
range,4,30.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0
elsevier,mary,324561,Job Position,1,50.000000,Educational
Qualification,1,20.000000,0,0,0,Work Experience,4,30,Aggregate
score,2,35.000000,Salary
range,4,35.000000,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0
```

hr_details_des.csv

```
symrise,kanan,376554,0,0,0,0,0,0,Educational Institution,1,15.000000,0,0,0,0,0,0,0,0,0,Number
of working hours,1,17.500000,Transportation,2,2.500000,Shift,3,17.500000,Work from
Home,3,15.000000,Employee Type,1,7.500000,Job Location,2,22.500000,Number of paid
leaves,1,17.500000,Stock options and incentives,1,15.000000,Medical Insurance for
family,1,10.000000
dragaco,prahalad,873452,0,0,0,0,0,0,Educational
Institution,1,15.000000,0,0,0,0,0,0,0,0,0,Number of working
hours,1,7.500000,Transportation,1,2.500000,Shift,3,17.500000,Work from
Home,3,15.000000,Employee Type,1,7.500000,Job Location,2,15.000000,Number of paid
leaves,1,17.500000,Stock options and incentives,1,15.000000,Medical Insurance for
family,1,10.000000
qualcom,Ganesh,432156,0,0,0,0,0,0,Educational
Institution,1,15.000000,0,0,0,0,0,0,0,0,0,Number of working
hours,2,17.500000,Transportation,1,2.500000,Shift,3,17.500000,Work from
Home,3,15.000000,Employee Type,1,7.500000,Job Location,2,22.500000,Number of paid
leaves,1,17.500000,Stock options and incentives,1,15.000000,Medical Insurance for
family,1,10.000000
elsevier,mary,324561,0,0,0,0,0,0,Educational Institution,2,15.000000,0,0,0,0,0,0,0,0,0,Number
of working hours,3,17.500000,Transportation,4,2.500000,Shift,2,17.500000,Work from
Home,3,15.000000,Employee Type,1,7.500000,Job Location,2,22.500000,Number of paid
leaves,1,17.500000,Stock options and incentives,1,15.000000,Medical Insurance for
family,1,10.000000
```

OPTION 4 - If choice = 4 is entered displaying the details of accepted vacancies only if the matching algorithm was initiated and suitable outcomes are displayed.

```
else if (choice2 == 4) {
  FILE *file5 = fopen("ap_ranks.csv", "r");
  struct hr_suggestions hr_suggestions[3];
  char name11[40], job11[3];
  // flag3 to navigate if profile found
  int flag3 = 0;
  printf("\nEnter name of applicant:");
  scanf("%s", name11);
  printf("\nEnter Job Position number:");
  scanf("%s", job11);
  char line[256];
  char *token;

  // Read each line from the file
  while (fgets(line, sizeof(line), file5)) {…}

  // Close the file
  fclose(file5);
  if (flag3 == 0) {
    printf("\nSorry matching not initiated!\n");
  }
}
```

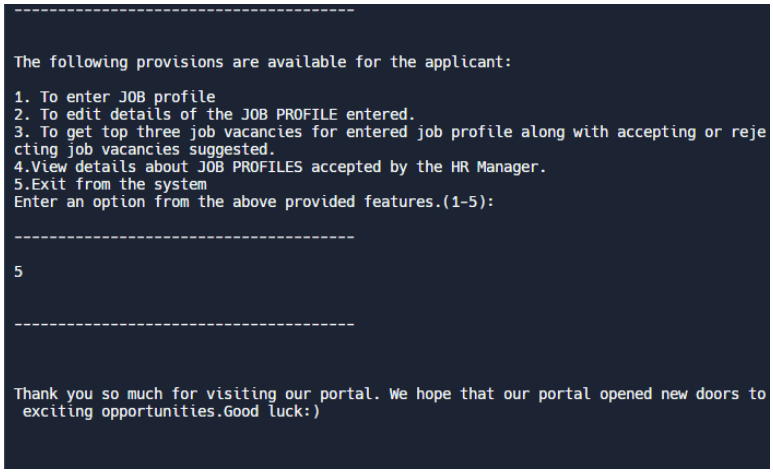
OUTPUT:

```
4.View details about JOB PROFILES accepted by the HR Manager.
5.Exit from the system
Enter an option from the above provided features.(1-5):

---------------------------------------

4

Enter name of applicant:vinayaka

Enter Job Position number:1

Details about the HR manager along with the JOB position entered and profiles accpeted

vinayaka
1
symrise
kanan
kanan@symrise.com
dragaco
prahalad
prahalad@dragaco.com



Welcome APPLICANT!
```

If applicant wishes to exit from the system:(choice 5)

```
else if (choice2 == 5) {
  printf(
      "\n\nThank you so much for visiting our portal. We hope that our "
      "portal opened new doors to exciting opportunities.Good "
      "luck:)\n\n");
  printf("\n\n----------------------------------------\n\n");
  return 1;
}
```

OUTPUT:



A validation is also entered incase of any stray user with undefined roles:

```
else {
  printf("\nRole not recognised,user not authorised!");
  printf("\nExiting system!");
  return 1;
}
```

Observations from the societal and legal perspectives:

Societal Perspective

- Mediating the supply demand between Job vacancies and Skilled labor in the market
- Providing a free portal to provide more opportunities to the deserving.
- Reducing job dissatisfaction and improving job retention as compulsory and desirable factors are given importance during algorithm matching.
- Following an unbiased, transparent recommendation process ensures equality and fairness.

Legal Perspective

- Ensuring safety and privacy of the data entered.
- Follow the Information Technology Act, 2000 as also rules, regulations, guidelines, bye laws and notifications made thereunder.
- Ethical use of the data entered.

Unique Implementations in the project:

1. Our platform focuses on reducing job dissatisfaction and enhancing job retention by prioritizing compulsory and desirable factors during the algorithmic matching process.
2. Inputs not only standard fields like academic details , but also incorporates fields like salary, night shift availability,Transportation(shuttle/bus service),Number of paid leaves per year,Flexible work schedule and availability of work from home facility ,Insurance coverage for family and Stock options or any other long-term incentives.
3. Applicants or HR can review their accepted options received from the outcome of the matching algorithm.
4. Also the user's role need not be mentioned during login as the system remembers the role associated with the login ID thus making the system more secure and away from stary users.
5. By providing a free portal, we strive to create more opportunities for

deserving individuals, promoting inclusivity and equal access to employment prospects.
6. Easy to access and enter value as it is mainly menu-driven
7. Gives very less scope for data inconsistency as the details are taken from the menu
8. We have kept in mind the features that can be scaled up complying with the Information Technology Act, 2000, along with all relevant rules, regulations, guidelines, bye-laws, and notifications established under this act.

Limitations of the solution provided

- Our platform currently does not support direct/indirect interaction between applicants and hiring managers. However, we are considering the addition of a communication feature to facilitate communication between these two parties in the future.
- Our platform caters to individuals in select fields or professions. Users can choose from the provided options to find relevant job opportunities that align with their expertise.
- The matching algorithm on our platform is based on the details entered by users. Currently, we do not offer background verification or validation services for the entered information.
- Our platform utilizes a file-based data storage method, which may pose technical challenges related to scalability, performance, and security. We are actively working on addressing these concerns to ensure a seamless user experience.

Learning Outcome

- Gained insights into the challenges and requirements of the HR recruitment process.
- Understanding the priorities of the current day job seekers who look for various factors other than just the job description and salary requirements.
- Developed skills in planning, setting deadlines, allocating resources, and monitoring progress using the replit platform that allowed us to monitor each other's progress.
- Ensuring usability and satisfaction for both the roles at least to a certain extent using the c programming language.
- Learned to communicate, coordinate tasks, and work together towards a common goal with people from different regional backgrounds
- Gained experience in working with data structures, file I/O operations, and other key concepts of C programming practically by coding and debugging the errors
- Learnt on how to create a basic architecture that involves designing and implementing various features and functionalities

References:

1. https://www.naukri.com/

2. https://in.linkedin.com/

3. https://eprocure.gov.in/cppp/rulesandprocs/kbadqkdlcswfjdelrquehwuxcfmijmuixngudufgbuubgubfugbububjxcgfvsbdihbgfGhdfgFHytyhRtMjk4NzY=

4. https://www.ijirt.org/master/publishedpaper/IJIRT142731_PAPER.pdf

5. https://www.geeksforgeeks.org/employee-record-system-in-c-using-file-handling/