GROUP 6 PRESENTS

JOBSEEKER'S COMPANION

Mirza Ananta Hermawan (2206822212)

Darrel Khayru Risyad (2206820781)

BACKGROUND *of the program*

HIGH VOLUME OF APPLICATIONS

Fresh graduates often **apply to numerous job applications** potentially hundreds or thousands during their job search which makes **tracking applications challenging without a systematic approach.**

NEED FOR ORGANIZATION

Job seekers, especially university students and recent graduates, **require a method to efficiently manage their job applications** to avoid missing deadlines, interviews, or follow-ups.

IMPORTANCE OF TIMELY FOLLOW-UPS

Keeping track of applications and timely follow-up can significantly increase the chances of securing a job. Missing out on these could lead to missed opportunities.

OBJECTIVE *of the program*

- **Streamline Management:** Simplify the management of job applications from a single platform.
- **Track Progress:** Monitor each step of the application process, from submission to final decision.
- **Custom Searches:** Quickly find applications by job title, company, date, or field.
- **Data Persistence:** Ensure application data is saved and retrievable across multiple sessions.
- **Support Decision Making:** Aid users in prioritizing and following up on applications effectively.

ALGORITHM: PSEUDOCODE

```
Include FP_06.h
Function main
    Initialize head to load_application("applications.csv")
    Declare integer variable pilihan
      Declare character arrays nama_pekerjaan, nama_perusahaan, tanggal_lamaran, status
    Loop Forever
        Display Main Menu
        Input choice into pilihan
        Switch
            Case 1:  // Add new job application
                Prompt and input job title into nama_pekerjaan
                Prompt and input company name into nama_perusahaan
                Prompt and input application date into tanggal_lamaran
                Prompt and input status into status
                Create new application using input data
                Append new application to linked list head
                Break
Case 2: // Show all applications
 Call show_application(head)
 Break
 Case 3: // Delete all applications
 Call delete_all_application(head)
 Break
 Case 4: // Save and exit
 Call save_application(head, "applications.csv")
 Display "Applications saved. Exiting..."
 Return 0 // Exit program
 Case 5: // Help
 Call display_help()
 Break
 Case 6: // Edit status of an application
 Call edit_status(head)
 Break
 Default:
 Display "Invalid choice, please try again."
 End Switch
 End Loop
 Return 0
```

THE PROGRAM

1

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>

typedef struct job_application {//defining the structure
    char nama_pekerjaan[100];
    char nama_perusahaan[100];
    char tanggal_lamaran[15];
    char status[50];
    struct job_application *next_struct;
} job_application;

job_application* create_application(char *nama_pekerjaan, char *nama_perusahaan, char *tanggal_lamaran, char *status) {
    job_application *new = (job_application*) malloc(sizeof(job_application));
    if (new == NULL) {
        printf("error allocating memory\n");
        exit(1);
    }//to create a new job_application structure
    strcpy(new->nama_pekerjaan, nama_pekerjaan);
    strcpy(new->nama_perusahaan, nama_perusahaan);
    strcpy(new->tanggal_lamaran, tanggal_lamaran);
    strcpy(new->status, status);//copying the existing data to the new structure
    new->next_struct = NULL;//setting the new pointer so it wouldn't be associated with the previous struture
    return new;
}//overall, this part is to have a new structure to move in data so the previous structure can receive input again- part of linked list.

job_application* append_application(job_application *head, job_application *new) {
    if (head == NULL) {
```

2

```c
job_application* append_application(job_application *head, job_application *new) {
    if (head == NULL) {
        return new;
    }
    job_application *cursor = head;
    while (cursor->next_struct != NULL) {
        cursor = cursor->next_struct;
    }
    cursor->next_struct = new;
    return head;
}//travesing the whole list to find the end of it to indentify if its empty or not.
 //if empty, then that point (next_struct) will be appended to be the new head.

void save_application(job_application *head, const char *filename) {
    FILE *file = fopen(filename, "w");
    if (file == NULL) {
        printf("error opening file!\n");
        return;
    }
    job_application *traverse1 = head;
    while (traverse1 != NULL) {
        fprintf(file, "%s,%s,%s,%s\n", traverse1->nama_pekerjaan, traverse1->nama_perusahaan, traverse1->tanggal_lamaran, traverse1->status);
        traverse1 = traverse1->next_struct;
    }
    fclose(file);
}//file handling (opening and closing the saved data)

job_application* load_application(const char *filename) {
    FILE *file = fopen(filename, "r");
    char line[256];
    job_application *head = NULL;
    job_application *new = NULL;
    char nama_pekerjaan[100], nama_perusahaan[100], tanggal_lamaran[15], status[50];

    if (file == NULL) {
        printf("error opening file!\n");
        return NULL;//says that there is no file to open/no file in regards to this program as it reads NULL
    }
```

THE PROGRAM

```
73          printf("error opening file!\n");
74          return NULL;//says that there is no file to open/no file in regards to this program as it reads NULL
75      }
76
77      while (fscanf(file, "%99[^,],%99[^,],%14[^,],%49[^\n]\n", nama_pekerjaan, nama_perusahaan, tanggal_lamaran, status) != EOF) {
78          new = create_application(nama_pekerjaan, nama_perusahaan, tanggal_lamaran, status);
79          head = append_application(head, new);
80      }//telling the code how the file want to be read before creating a structure to be appended with said data and pointing "head" to the start of the list
81      fclose(file);
82      return head;
83  }
84
85  void tolower_String(char *str) {
86      int i = 0;
87      for (i; str[i]; i++) {
88          str[i] = tolower(str[i]);
89      }
90  }//making the strings to be not case sensitive
91
92  int convert_date_to_integer(const char *tanggal_lamaran) {
93      int day, month, year;
94      sscanf(tanggal_lamaran, "%d/%d/%d", &day, &month, &year);
95      return year * 10000 + month * 100 + day;
96  }//changing the date into numbers to be sorted
97
98  void bubble_sort_by_date(job_application **head) {
99      int swapped;
100     job_application *ptr1;
101     job_application *ptr2 = NULL;
102
103     if (*head == NULL) return;
104
105     do {
106         swapped = 0;
107         ptr1 = *head;
108
109         while (ptr1->next_struct != ptr2) {
110             if (convert_date_to_integer(ptr1->tanggal_lamaran) > convert_date_to_integer(ptr1->next_struct->tanggal_lamaran)) {
111                 // Swap the data of the nodes
112                 //Swapping and comparison by dates as it compares the integer form pointed by ptr1 and next_struct. if it's bigger, then the swapping occurs
113                 char temp_nama_pekerjaan[100], temp_nama_perusahaan[100], temp_tanggal_lamaran[15], temp_status[50];
114                 //dummy/temporary variables
115                 strcpy(temp_nama_pekerjaan, ptr1->nama_pekerjaan);
116                 strcpy(temp_nama_perusahaan, ptr1->nama_perusahaan);
117                 strcpy(temp_tanggal_lamaran, ptr1->tanggal_lamaran);
118                 strcpy(temp_status, ptr1->status);//segment: copying ptr1 data to temporary variable
119
120                 strcpy(ptr1->nama_pekerjaan, ptr1->next_struct->nama_pekerjaan);
121                 strcpy(ptr1->nama_perusahaan, ptr1->next_struct->nama_perusahaan);
122                 strcpy(ptr1->tanggal_lamaran, ptr1->next_struct->tanggal_lamaran);
123                 strcpy(ptr1->status, ptr1->next_struct->status);//segment: copying next_struct data to ptr1
124
125                 strcpy(ptr1->next_struct->nama_pekerjaan, temp_nama_pekerjaan);
126                 strcpy(ptr1->next_struct->nama_perusahaan, temp_nama_perusahaan);
127                 strcpy(ptr1->next_struct->tanggal_lamaran, temp_tanggal_lamaran);
128                 strcpy(ptr1->next_struct->status, temp_status);//segment: copying temporary variable data to next_struct
129
130                 swapped = 1;//
131             }
132             ptr1 = ptr1->next_struct;
133         }
134         ptr2 = ptr1;
135     } while (swapped);
136 }
137
138 void search_application(job_application *head, const char *keyword) {
139     job_application *traverse2 = head;
140     char searching[100];
141     strcpy(searching, keyword);
142     tolower_String(searching);
143     //to display the result/data
144     printf("\nSearch results:\n");
145     printf("-------------------------------------------------------------------------------\n");
146     printf("|%-30s|%-20s|%-10s|%-15s|\n", "Job title", "Company name", "Date", "Status");
147     printf("-------------------------------------------------------------------------------\n");
```

3

4

THE PROGRAM

```c
    printf("-------------------------------------------------------------------------------\n");
    printf("|%-30s|%-20s|%-10s|%-15s|\n", "Job title", "Company name", "Date", "Status");
    printf("-------------------------------------------------------------------------------\n");

    while (traverse2 != NULL) { //a pointer to go to each node of the list
        char pekerjaan_lower[100], perusahaan_lower[100];
        strcpy(pekerjaan_lower, traverse2->nama_pekerjaan);//copying the data in nama_pekerjaan to pekerjaan_lower
        strcpy(perusahaan_lower, traverse2->nama_perusahaan);//copying the data in nama_pekerjaan to pekerjaan_lower
        tolower_String(pekerjaan_lower);//turning the data to lower case
        tolower_String(perusahaan_lower);//turning the data to lower case

        if (strstr(pekerjaan_lower, searching) != NULL || strstr(perusahaan_lower, searching) != NULL) {
            printf("|%-30s|%-20s|%-10s|%-15s|\n", traverse2->nama_pekerjaan, traverse2->nama_perusahaan, traverse2->tanggal_lamaran, traverse2->status);
        }//if the data sent is not null, then it will print the result.
        traverse2 = traverse2->next_struct;
    }
    printf("-------------------------------------------------------------------------------\n");
}

void print_application(job_application *head) {
    job_application *traverse1 = head;
    int index = 1;
    printf("\nCurrent job applications:\n");
    printf("-----------------------------------------------------------------------------------------------\n");
    printf("| %-3s | %-30s | %-20s | %-10s | %-15s |\n", "No", "Job title", "Company name", "Date", "Status");
    printf("-----------------------------------------------------------------------------------------------\n");
    while (traverse1 != NULL) {
        printf("| %-3d | %-30s | %-20s | %-10s | %-15s |\n", index++, traverse1->nama_pekerjaan, traverse1->nama_perusahaan, traverse1->tanggal_lamaran, traverse1->status);
        traverse1 = traverse1->next_struct;
    }
    printf("-----------------------------------------------------------------------------------------------\n");
}//displaying the applications/tracker

void display_help(void) {
    printf("Help menu:\n");
    printf("1. Add new job application = add new job application\n");
    printf("2. Show all applications = display all recorded job applications\n");
    printf("3. Delete all applications = remove all job applications from the list\n");
    printf("4. Save & Exit = save all changes and exit the program\n");
    printf("5. Help = display this help menu\n");
    printf("6. Edit status application = edit the status of an existing job application\n");
}//display the "help" part just in case someone need some explanation

void delete_all_application(job_application **head) {
    job_application *traverse1 = *head;//initialize traverse1 to the head
    job_application *next = NULL;

    while (traverse1 != NULL) {
        next = traverse1->next_struct;//next is set to point to to the next node
        free(traverse1);//freeing memomry
        traverse1 = next;
    }
    *head = NULL;
    printf("All job applications have been deleted.\n");
}

void show_application(job_application *head) {//function whether to show or search data
    int choice;
    while (1) {
        printf("\n1. Sort by Date\n2. Search by Keyword\n3. Return\nEnter your choice: ");
        scanf("%d", &choice);
        getchar();  // consume newline

        switch (choice) {
            case 1:
                bubble_sort_by_date(&head);
                print_application(head);
                break;
            case 2:
                printf("Enter search keyword (job title or company name): ");
                char keyword[100];
                scanf(" %[^\n]", keyword);
                search_application(head, keyword);
                break;
            case 3:
                return;
            default:
                printf("Invalid choice, please try again.\n");
        }
    }
}

void edit_status(job_application *head) {
    if (head == NULL) {
        printf("No job applications available to edit.\n");
        return;//if there's not data in the first place
    }

    print_application(head);

    int index, count = 1;
    job_application *traverse2 = head;//setting the traverse to the head of the list
    printf("Enter the number of the application you want to edit: ");
    scanf("%d", &index);
    getchar();  // consume newline

    while (traverse2 != NULL && count < index) {
        traverse2 = traverse2->next_struct;//setting traverse to point to next node
        count++;
    }

    if (traverse2 == NULL) {
        printf("Invalid application number.\n");
        return;
    }

    printf("Current status: %s\n", traverse2->status);
    printf("Enter new status: ");
    scanf(" %[^\n]", traverse2->status);//changing the status
    printf("Status updated successfully.\n");
}
```

OUTPUT OF THE PROGRAM

```
Enter your choice: 2

1. Sort by Date
2. Search by Keyword
3. Return
Enter your choice: 1

Current job applications:
-----------------------------------------------------------------------------------------
| No  | Job title                 | Company name          | Date       | Status          |
-----------------------------------------------------------------------------------------
| 1   | Director                  | Samsung               | 20/02/2022 | on progress     |
| 2   | Admin                     | Intel                 | 20/03/2022 | accepted        |
-----------------------------------------------------------------------------------------
```

```
Enter search keyword (job title or company name): samsung

Search results:
-----------------------------------------------------------------------------------------
|Job title                  |Company name           |Date       |Status           |
-----------------------------------------------------------------------------------------
|Director                   |Samsung                |20/02/2022|on progress      |
-----------------------------------------------------------------------------------------
```

```
-----------------------------------------------------------------------------------------
Enter the number of the application you want to edit: 1
Current status: on progress
Enter new status: accepted
Status updated successfully.

====================================
Welcome to the JobSeeker's Companion
====================================
1. Add new job application
2. Show all applications
3. Delete all applications
4. Save and exit
5. Help
6. Edit status application
====================================
Enter your choice: 2

1. Sort by Date
2. Search by Keyword
3. Return
Enter your choice: 1

Current job applications:
-----------------------------------------------------------------------------------------
| No  | Job title                 | Company name          | Date       | Status          |
-----------------------------------------------------------------------------------------
| 1   | Director                  | Samsung               | 20/02/2022 | accepted        |
| 2   | Admin                     | Intel                 | 20/03/2022 | accepted        |
```

```
Enter your choice: 3
All job applications have been deleted.

====================================
Welcome to the JobSeeker's Companion
====================================
1. Add new job application
2. Show all applications
3. Delete all applications
4. Save and exit
5. Help
6. Edit status application
====================================
Enter your choice: 2

1. Sort by Date
2. Search by Keyword
3. Return
Enter your choice: 1

Current job applications:
-----------------------------------------------------------------------------------------
| No  | Job title                 | Company name          | Date       | Status          |
-----------------------------------------------------------------------------------------
```

THANK You

Mirza Ananta Hermawan (2206822212)

Darrel Khayru Risyad (2206820781)