

EMPLOYEE MANAGEMENT

PROJECT REPORT

Submitted by

DEVI R	717821E114
DIVYASHREE Y	717821E116
GOPIKA K L	717821E119
REEMA MANSOORA S	717821E143
SURYAVADHANI D	717821E155

21EC09 DATA STRUCTURES (SEMESTER 4)



KARPAGAM COLLEGE OF ENGINEERING

(Autonomous)

COIMBATORE – 641 032

MAY 2023



CERTIFICATE

Certified that this project report titled “**Employee Management**” is the bonafide work of **DEVI R (717821E114), DIVYASHREE Y (717821E116), GOPIKA K L (717821E119), REEMA MANSOORA S (717821E143), SURYAVADHANI D (717821E155)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report.

I understand the policy on plagiarism and declare that the project and publications are my own work, except where specifically acknowledged and has not been copied from other sources or been previously submitted for award or assessment.

[Mr.N.VARATHARAJAN]
Faculty In-charge

Submitted for the Project Viva-Voce examination held on _____

Examiner I

Examiner II

ABSTRACT

Employers could gain insight into their employees via an “Employee management system”, which allows them to better plan and manages work hours, lowering labor expenses and increasing productivity.

This report includes a development presentation of an information system for managing the staff data within a small company or organization. The system as such as it has been developed is called Employee Management System. It consists of functionally related GUI (application program) and database. The choice of the programming tools is individual and particular.

In this series of C Projects Source Code, we'll look at how to build an Employee Management System in C. We may manage the information of workers working in a firm or organization using this Employee Management System. The file handling technique is used here to save the data in a particular file, and you get the notion of this project as soon as you hear the name.

This project uses the Insert, Edit, and Delete file actions, but the sole constraint is that you can only display the data, not search for any data item in particular. If you have more experience with C, you may alter this program by using the searching strategies.

TABLE OF CONTENTS

S.NO.	TITLE	PAGE NO.
	LIST OF FIGURES	5
1.	PROBLEM DEFINITION	6
2.	REQUIREMENTS SPECIFICATION	7
	2.1 HARDWARE REQUIREMENTS	
	2.2 SOFTWARE REQUIREMENTS	
3.	DATA STRUCTURES USED	8
4.	ALGORITHM DESIGN	9
	4.1 ALGORITHM	
5.	SOURCE CODE	10
6.	RESULTS & DISCUSSION	17
7.	CONCLUSION	20
8.	BIBLIOGRAPHY	21

LIST OF FIGURES

S.NO	FIGURE NAME	PAGE NO
1.	fig 6.1: First establishing screen after running the program	17
2.	fig 6.2: The option that are available in the project	17
3.	fig 6.3: Information to be asked for adding data	18
4.	fig 6.4: Information to be asked for deleting data	18
5.	fig 6.5: Process of exit	19

1 PROBLEM DEFINITION

The employee record system is based on the menu-driven program. It consists of following features.

- Writing the data in binary file
- Reading the data from binary file
- Modify the record
- Delete the record

Use appropriate data structure to implement this.

2 REQUIREMENTS SPECIFICATION

2.1 HARDWARE REQUIREMENTS Processor: Core i3/ i5/ i7 RAM: 128 MB

Hard Disk: 20GB

Monitor: 15” Color monitor Key Board: 104 Keys

2.2 SOFTWARE REQUIREMENTS Operating System: Windows7/8. Language: C

Software: DevC++

3 DATA STRUCTURES USED

The main data structure used in this code is a struct named "Emp". It has four fields: "name", "salary", "age" and "id", which are used to store information about employee's name, salary, age and id of employee respectively. An array of type "empname" is used to store multiple emp objects. This array is created using the user input of the number of employee's information to be added, and it is later passed as an argument to the sorting functions.

- 1.Include necessary libraries - `stdio.h`, `stdlib.h`, `string.h`, `windows.h`.
- 2.Define an Employee struct that consists of four fields - name (a string of maximum 50 characters), salary (a float), age (an integer), and id (an integer).
- 3.Storing the details like name of the employee, ID, salary, age.
- 4.Adding new employee details to the existing list.
- 5.Define four functions-add records, delete records, display records, modify records.
- 6.These functions help the user to make any changes.
- 7.Define a switch case for the four functions.
- 8.Add a new employee's details.
- 9.Update the details of an existing employee.
- 10.Delete (fired from the company) the details of a specific employee from the database.
- 11.Define the while case so that it will print all the details of employees.
- 12.Return 0 to indicate successful program execution.

```

// C program for the above approach
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <windows.h>

// Structure of the employee
struct emp {
    char name[50];
    float salary;
    int age;
    int id;
};
struct emp e;

// size of the structure
long int size = sizeof(e);

// In the start coordinates
// will be 0, 0
COORD cord = { 0, 0 };

// function to set the
// coordinates
void gotoxy(int x, int y)
{
    cord.X = x;
    cord.Y = y;
    SetConsoleCursorPosition(
        GetStdHandle(STD_OUTPUT_HANDLE),
        cord);
}

FILE *fp, *ft;

// Function to add the records
void addrecord()
{
    system("cls");
    fseek(fp, 0, SEEK_END);
    char another = 'y';

```

```

while (another == 'y') {
    printf("\nEnter Name : ");
    scanf("%s", e.name);

    printf("\nEnter Age : ");
    scanf("%d", &e.age);

    printf("\nEnter Salary : ");
    scanf("%f", &e.salary);

    printf("\nEnter EMP-ID : ");
    scanf("%d", &e.id);

    fwrite(&e, size, 1, fp);

    printf("\nWant to add another"
           " record (Y/N) : ");
    fflush(stdin);

    scanf("%c", &another);
}
}

```

```

// Function to delete the records
void deleterecord()
{
    system("cls");
    char empname[50];
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter employee "
               "name to delete : ");
        scanf("%s", empname);

        ft = fopen("temp.txt", "wb");
        rewind(fp);

        while (fread(&e, size,
                     1, fp)
               == 1) {
            if (strcmp(e.name,
                      empname)

```

```

        != 0)
        fwrite(&e, size, 1, ft);
    }

    fclose(fp);
    fclose(ft);
    remove("data.txt");
    rename("temp.txt", "data.txt");
    fp = fopen("data.txt", "rb+");

    printf("\nWant to delete another"
           " record (Y/N) :");
    fflush(stdin);
    another = getche();
}
}

// Function to display the record
void displayrecord()
{
    system("cls");

    // sets pointer to start
    // of the file
    rewind(fp);

    printf("\n=====
           "=====
           "=====");
    printf("\nNAME\t\tAGE\t\tSALARY\t\t"
           "\tID\n",
           e.name, e.age,
           e.salary, e.id);
    printf("=====
           "=====
           "=====\n");

    while (fread(&e, size, 1, fp) == 1)
        printf("\n%s\t\t%d\t\t%.2f\t\t10d",
               e.name, e.age, e.salary, e.id);

    printf("\n\n\n\t");
    system("pause");
}

```

```

// Function to modify the record
void modifyrecord()
{
    system("cls");
    char empname[50];
    char another = 'y';

    while (another == 'y') {
        printf("\nEnter employee name"
            " to modify : ");
        scanf("%s", empname);

        rewind(fp);

        // While File is open
        while (fread(&e, size, 1, fp) == 1) {
            // Compare the employee name
            // with ename
            if (strcmp(e.name, empname) == 0) {
                printf("\nEnter new name:");
                scanf("%s", e.name);
                printf("\nEnter new age :");
                scanf("%d", &e.age);
                printf("\nEnter new salary :");
                scanf("%f", &e.salary);
                printf("\nEnter new EMP-ID :");
                scanf("%d", &e.id);

                fseek(fp, -size, SEEK_CUR);
                fwrite(&e, size, 1, fp);
                break;
            }
        }

        // Ask for modifying another record
        printf("\nWant to modify another"
            " record (Y/N) :");
        fflush(stdin);
        scanf("%c", &another);
    }
}

// Driver code

```

```
int main()
{
    int choice;


// opening the file
fp = fopen("data.txt", "rb+");



// showing error if file is
// unable to open.
if (fp == NULL) {
    fp = fopen("data.txt", "wb+");
    if (fp == NULL) {
        printf("\nCannot open file...");
        exit(1);
    }
}




system("Color 3F");
printf("\n\n\n\t\t\t=====
      =====")
      "=====");
printf("\n\t\t\t~~~~~"
       "~~~~~"
       "~~~");
printf("\n\t\t\t\t=====
      =====")
      "=====");
printf("\n\t\t\t[|::>::~>::>:> <"
       "EMPLOYEE RECORD <::<::<:"
       "<:::]\\t");
printf("\n\t\t\t\t=====
      =====")
      "=====");
printf("\n\t\t\t~~~~~"
       "~~~~~"
       "~");
printf("\n\t\t\t\t=====
      =====\n");
printf("\n\n\n\t\t\t\t\t\t\t\t\t"
       "Project by EEE A Team 2"
       "\n\n\t\t\t\t\t");


system("pause");
```

```

while (1) {
    // Clearing console and asking the
    // user for input
    system("cls");
    gotoxy(30, 10);
    printf("\n1. ADD RECORD\n");
    gotoxy(30, 12);
    printf("\n2. DELETE RECORD\n");
    gotoxy(30, 14);
    printf("\n3. DISPLAY RECORDS\n");
    gotoxy(30, 16);
    printf("\n4. MODIFY RECORD\n");
    gotoxy(30, 18);
    printf("\n5. EXIT\n");
    gotoxy(30, 20);
    printf("\nENTER YOUR CHOICE...\n");
    fflush(stdin);
    scanf("%d", &choice);

    // Switch Case
    switch (choice) {
        case 1:

            // Add the records
            addrecord();
            break;
        case 2:
            // Delete the records
            deleterecord();
            break;
        case 3:
            // Display the records
            displayrecord();
            break;
        case 4:
            // Modify the records
            modifyrecord();
            break;
        case 5:
            fclose(fp);
            exit(0);
            break;

        default:

```

```
        printf("\nINVALID CHOICE...\n");
    }
}

return 0;
}
```


6 RESULT AND DISCUSSION

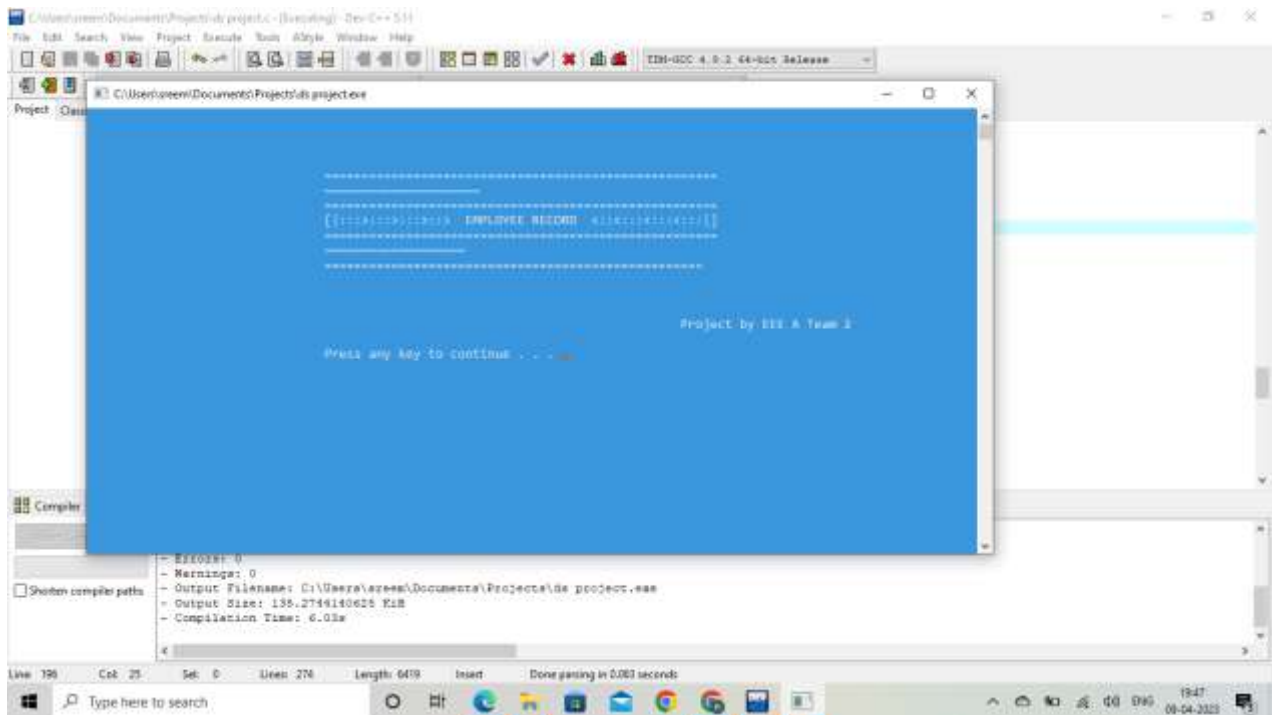


fig 6.1: First establishing screen after running the program

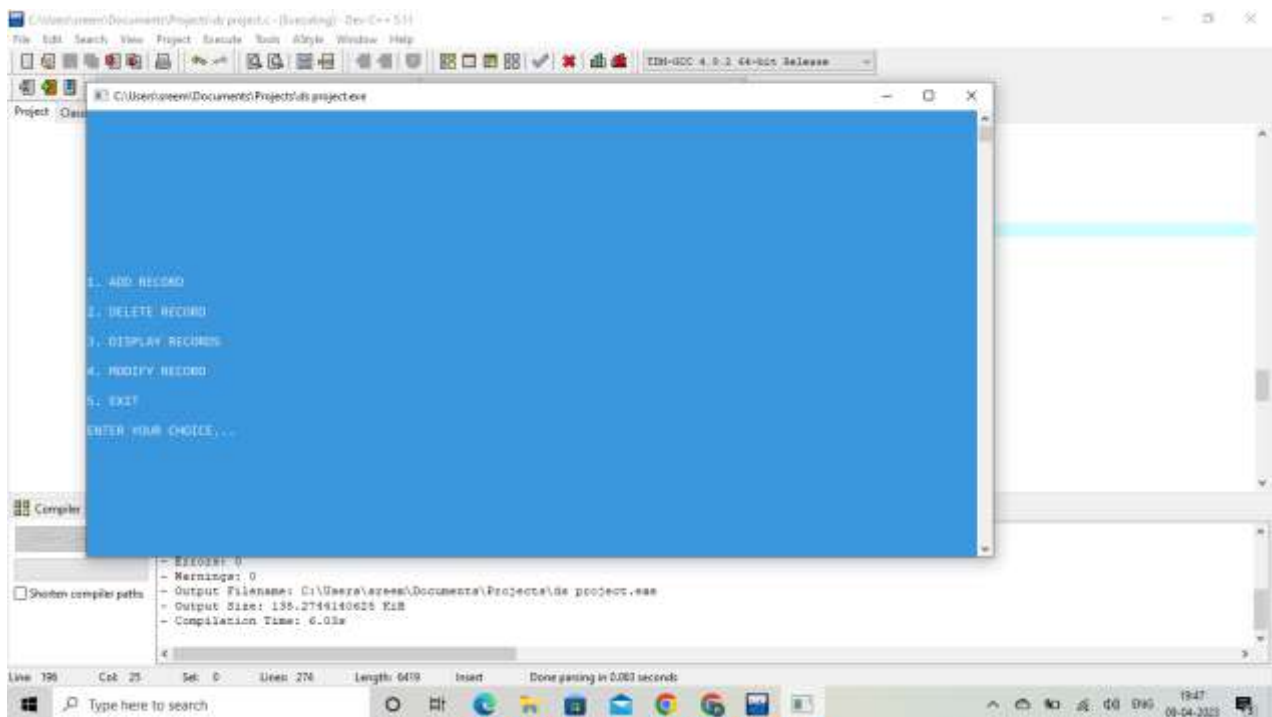


fig 6.2: The option that are available in the project

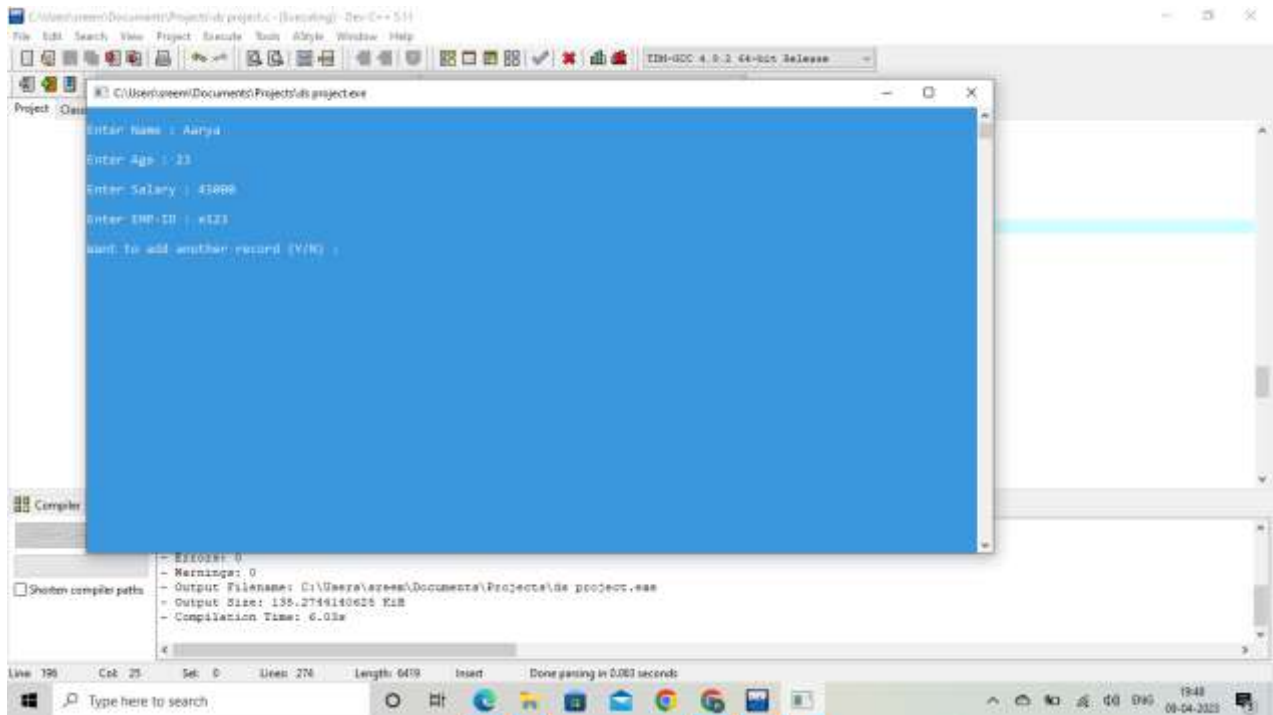


fig 6.3: Information to be asked for adding data

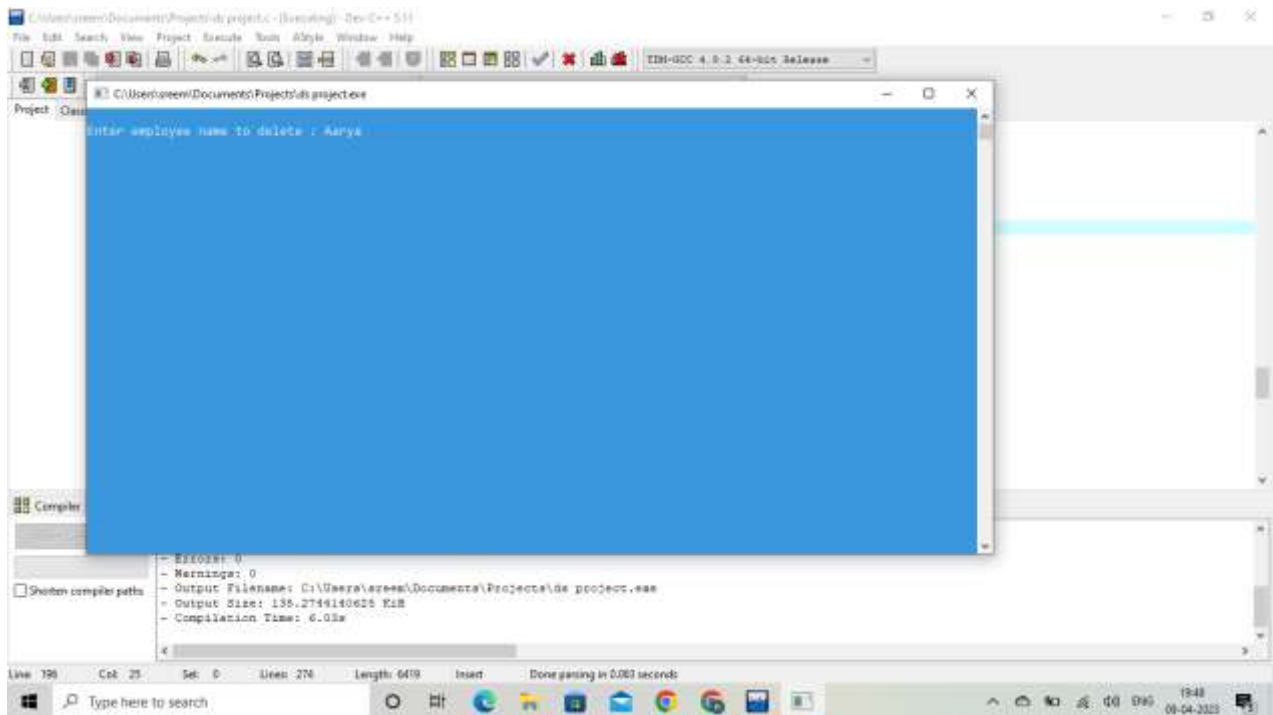


fig 6.4: Information to be asked for deleting data

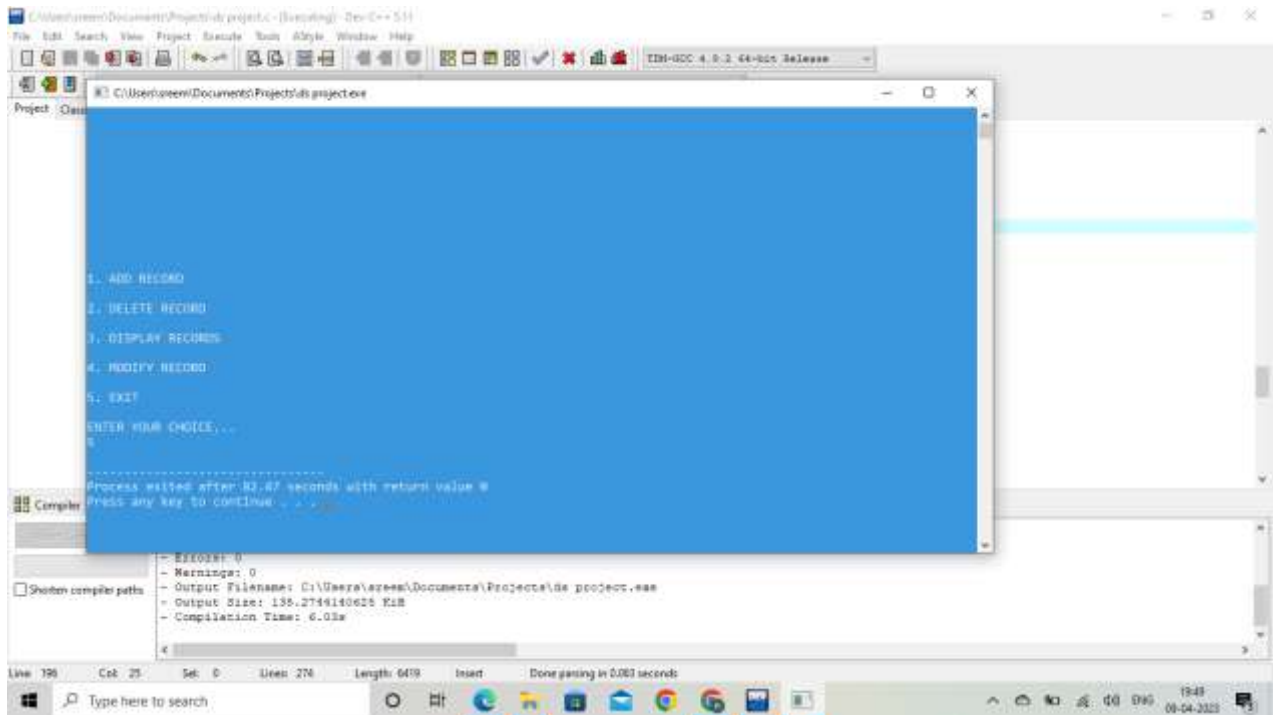


fig 6.5: Process of exit

In the above output the information of the employee is given by the user, it is then it is deleted or modified or displayed as per the user wish.

The employee management system project is proposed to effectively understand the work, the type of person who is fit for the job, and the organization. It empowers the employee to accomplish the job and manages employees very well..The program successfully completed and it may be added,deleted,displayed or modified as per the user need.

8 BIBLIOGRAPHIES

- 1) Srimulyani, Veronika Agustini, and Yustinus Budi Hermanto. "Employer branding and employee performance at KAI: employee retention' role as mediator." *Jurnal Studi Komunikasi (Indonesian Journal of Communications Studies)* 6, no. 3 (November 21, 2022): 921–40. <http://dx.doi.org/10.25139/jsk.v6i3.5381>.
- 2) D, VENKATRAMA RAJU, KAMARAJ S, and KANNAN J. "Employee Retention Techniques." *Journal of Research on the Lepidoptera* 50, no. 1 (March 25, 2019): 40-47. <http://dx.doi.org/10.36872/lepi/v50i1/201055>.
- 3) V.B. DEVIBALA, V. B. DEVIBALA, Dr P. ANBUOLI Dr.P.ANBUOLI, and T. JOTHIMURUGAN T.JOTHIMURUGAN. "Reason for Employee Retention." *Indian Journal of Applied Research* 4, no. 4 (October 1, 2011): 79–81. <http://dx.doi.org/10.15373/2249555x/apr2014/230>.