INTERNSHIP

PROJECT-1

EMPLOYEE MANAGEMENT SYSTEM

Contents :

Developed by: K.V.J.RAMAN

GUIDE: Mr. BHARATH

Mr. PRITHVI

Introduction

Hence after the completion of the project we got familiar with the C programming and its features. A complete and useful library management can only be developed with lot of intensive effort and time. Due to lack of time and we are beginners in programming program that we expected can’t be developed by us. Our library management may not be must useful for library in our college but it will be the most useful for study and programming practice using C. As a whole, the project has been a good learning experience for us. We have gained knowledge about the various aspects of C programming. At the same time, we have developed a deep understanding about the file handling in C.

Description

Employee management system is a project that helps us to store the employee information.Once the user login in this application so that we can add,delete,.save,browse the data.This project was implemented in C programming.

High level requirements

- It describes the set of capabilities in which a project must achieve a set of capabilities and the expected outcomes and this has to be delivered by the project.

- First of all we should have a clear idea about the project and a particular attention to the capabilities and conditions.

The main constraints for the high level requirements are:

- Windows(Operating System).

- C Language.

- Design.

- Implementation.

Low level Requirements

- Processing.

- Technical details.

- Functionalities.

- Calculations.

- Performance.

# 4W's & 1H

Who

- Employee Management System

What

- Employee management system helps us to store and to know about the data given by the employee.

When

- It helps us when the employee join in company or in any organization.

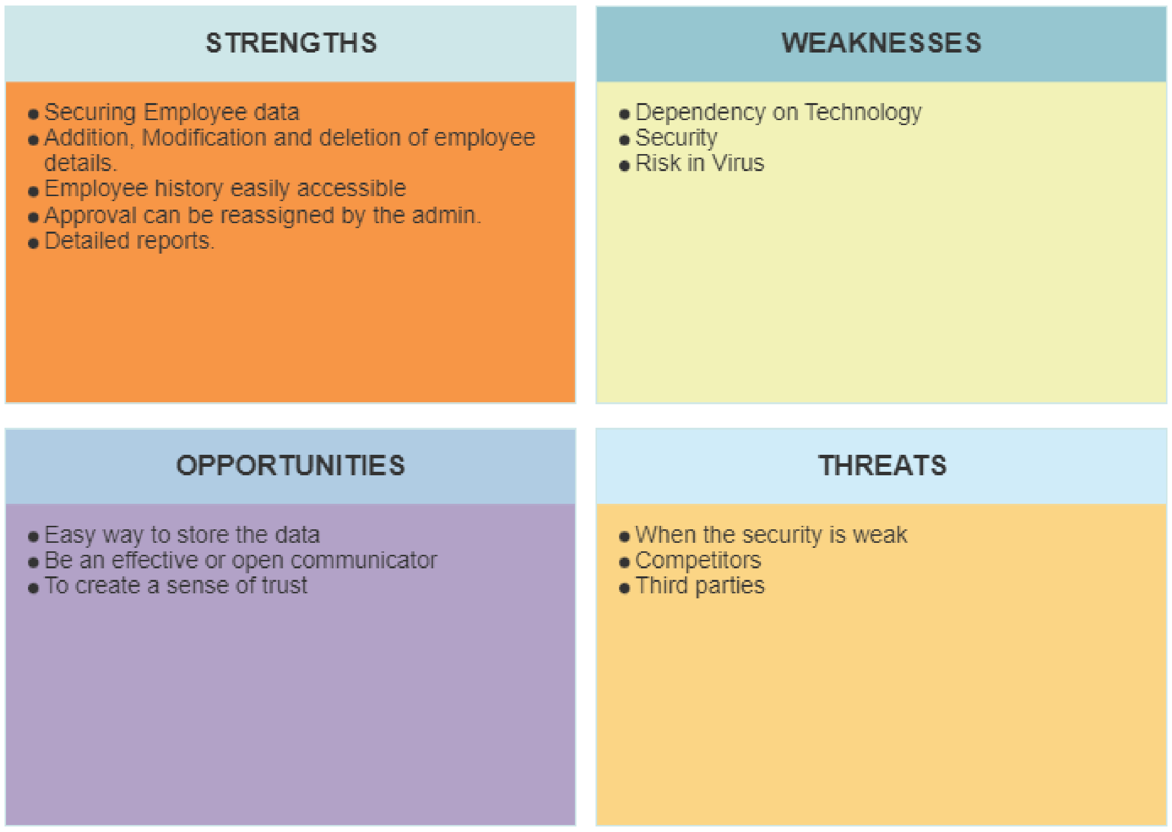
Where

- This is used in any company or organization.

How

- It is implemented in c language and by creating a system which will provide required details of an employee in a system of the company or the organization.

SWOT ANALYSIS



STRENGTHS

- Securing Employee data
- Addition, Modification and deletion of employee details.
- Employee history easily accessible
- Approval can be reassigned by the admin.
- Detailed reports.

WEAKNESSES

- Dependency on Technology
- Security
- Risk in Virus

OPPORTUNITIES

- Easy way to store the data
- Be an effective or open communicator
- To create a sense of trust

THREATS

- When the security is weak
- Competitors
- Third parties

Modules Description

Analysis: This system, in order to facilitate customer input and make the program clear, so I use the module method,
Modularize each function to make the calls between functions in the program clearer.

Function module description

Describe the functions of each module.

Entry module： Enter the information of the existing employees of the factory and save it in the file to facilitate the sorting and update
New, statistics, printing and other operations.

Output module: Output all employee information or required employee information on the screen for printing or checking
Ask for employee information.

Delete module: Delete employees who have retired or left the factory.

Add module: Add and save the new employee information.

Modify the module: Find the employee and select the option to be modified.

Sorting module: This program only sorts the age of employees from small to large.

Query module: This module is divided into query by name, age, and job.

Complete Source Code

```
#include<stdio.h>

#include<stdlib.h>

#include<string.h>

struct employee

{

        char no[40];   //Employee ID

        char name[40]; //Employee name

        char sex[10];  //Employee gender

        int age;        //Employee age

        struct employee *next;//Next node pointer

};

typedef struct employee EMP;

EMP *h;



void load();       //Read file data and create a linked list

void save();       //Write the data in the linked list to the file
void add(EMP *p);  //Add employee nodes in the linked list

void del(char *s); //Delete the employee node with the specified
name in the linked list

void update();    //Update the data in the employee node in the
linked list

void browse();    //Display all node data in the linked list on the
screen

void menu();      //Display system main menu

void delEmp();    //Delete employee

void addEmp();    //Add staff

void main(void)

{



        int op;//operation

        h =(EMP *)malloc(sizeof(EMP));   //Create the head node of
the linked list

        h->next = NULL;             //Initialize the next node pointer in
the head node of the linked list to NULL

        menu();
```

```
scanf("%d",&op);
    while(op!=0)
    {
           switch(op)
           {
             case 1:
                         load();
                         break;
             case 2:
                         save();
                         break;
             case 3:
                         addEmp();
                         break;
             case 4:
                         delEmp();
                         break;
```

```
                case 5:
                        update();
                        break;
                case 6:
                        browse();
                        break;
               }
          menu();
         scanf("%d",&op);
          }
}


//Display system main menu function
void menu()
{
    printf("        Main   Menu\n");
         printf("   **********************   \n");
```

```
          printf("   *  0:quit   1 :load   *  \n");

          printf("   *  2:save   3 :add    *  \n");

          printf("   *  4:delete 5 :update *  \n");

          printf("   *  6:browse 7 :default*  \n");

          printf("   **********************   \n");

}



//Read file data and create linked list function

void load()

{

  FILE *fp;

  char ch;



  fp = fopen("data.txt","r");

  if(fp==NULL)

  {
        printf("Error opening file, press any to return to the main menu\n");

        getch();//The program stops and waits for input

        return;

  }

  while(!feof(fp))

  {

        EMP *node =(EMP *)malloc(sizeof(EMP));

        node->next = NULL;

        fscanf(fp,"%s%s%s%d",node->no,node->name,node->sex,&node->age);//Read a line of employee information from the file

        add(node); // Call function to increase linked list node

  }

  fclose(fp);

}



//Function to add employee nodes in the linked list
```

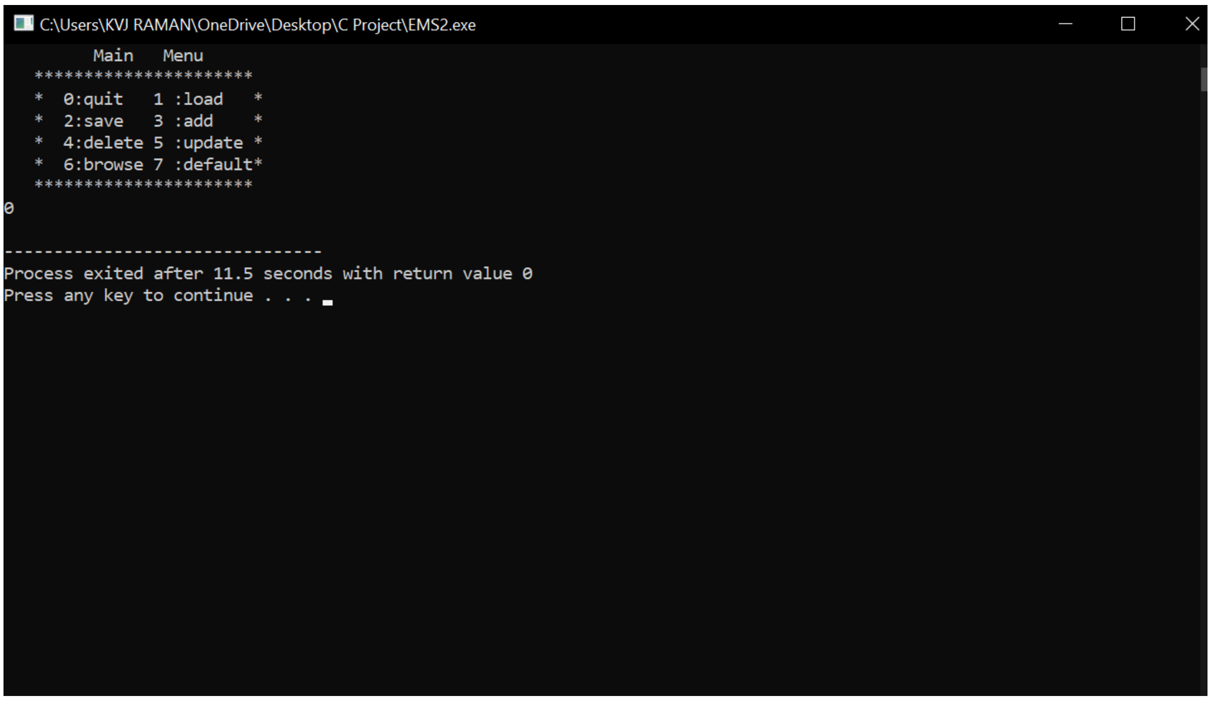
```
void add(EMP *p)
{
   EMP *q;
   q = h->next;  //Point the q pointer to the first employee node in
the linked list
   if(q==NULL)  //If the q pointer is NULL, it means that the
current linked list is empty
   {
            h->next = p; //The node pointed to by the p pointer is the
first node in the linked list
            p->next = NULL;
   }
   else
   {
            while(q->next!=NULL) //Find the last node in the linked
list through the while loop
            {
                    q = q ->next;
            }
            q->next = p;    //Point the next pointer of the last node
pointed to by the q pointer to the new node

            p->next =NULL;  //Now the p pointer points to the last
node, so the next pointer of the node is set to NULL

   }



}



//Add employee function

void addEmp()

{

  EMP *node;

  node = (EMP *)malloc(sizeof(EMP));

  node->next = NULL;

  printf("input the employee's no name sex age:\n");

  scanf("%s%s%s%d",node->no,node->name,node->sex,&node-
>age);

  add(node);
```

```
}

//Delete employee function
void delEmp()
{
   char name[40];
   printf("input del name:\n");
   scanf("%s",name);
   del(name);
}

//Display the data function of all employee nodes in the linked list
on the screen
void browse()
{
         EMP *node;
```

```
    node = h->next;  //node pointer points to the first node of the
linked list

         printf("Employee no    name  age  sex\n");

         while(node)   //Traverse the linked list

         {

                  printf("%12s%6s%5d%4s\n",node->no,node-
>name,node->age,node->sex);

        node = node->next;  //node pointer points to the next
employee node

         }

         printf("\n");

}




//The function to write the data in the linked list to the file

void save()

{

         FILE *fp;
        EMP *node;

        fp = fopen("data.txt","w"); //Open the file by writing

        node = h->next;     //node pointer points to the first node of
the linked list

        while(node!=NULL)

        {   //"%s %s %s %d\n" there are spaces between format
characters

                fprintf(fp,"%s %s %s %d\n",node->no,node-
>name,node->sex,node->age);

                node  = node ->next;

        }

        fclose(fp);

}



//Function to delete employee nodes

void del(char *s)

{

   EMP *p,*q;
```

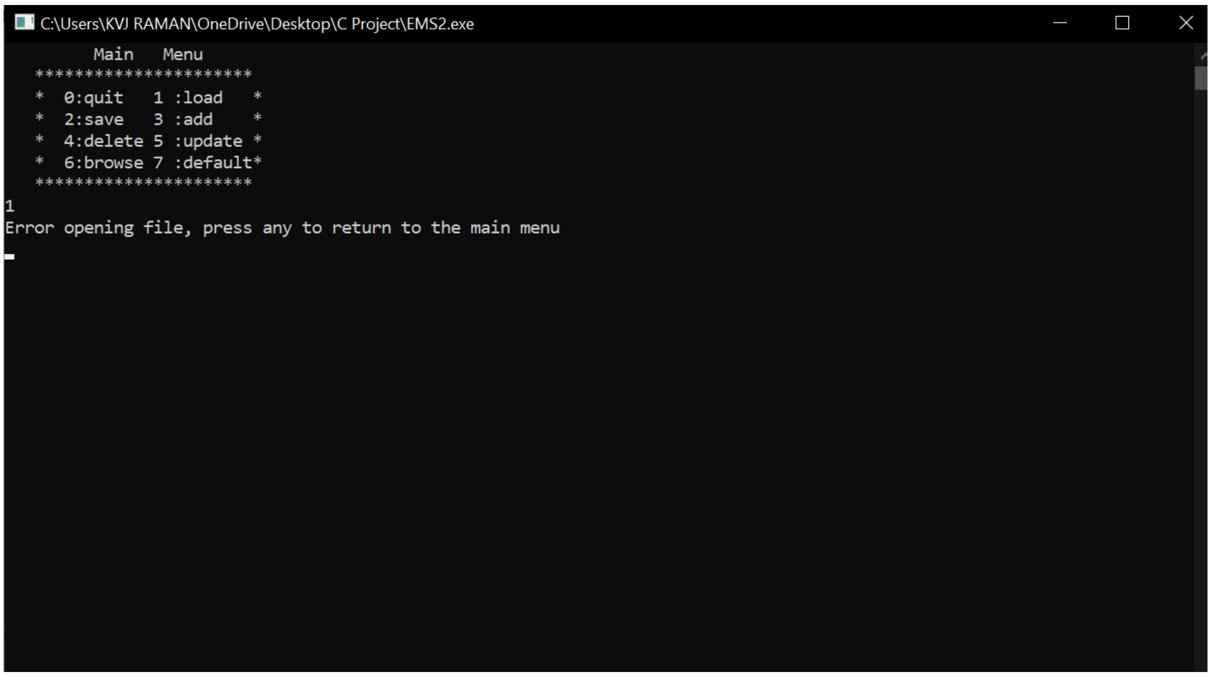
```
        p = h->next;//The first node in the linked list p->next=p->next->next;

        q = p->next;//The second node in the linked list

        if(strcmp(s,p->name)==0)

        {h->next=p->next;

         free(p);

         return;

        }

        while(q!=NULL)

        {

     if(strcmp(s,q->name)==0)

          {

       p->next = q->next;

                 free(q);

                 break;

          }

          p = q;
          q = q->next;

        }

}



//Update the function of the employee node

void update()

{

    char name[40];

         int flag = 0;

         EMP *p;

         printf("input the name:\n");

         scanf("%s",name);

    p = h->next;

         while(p!=NULL)

         {

                  if(strcmp(name,p->name)==0)

                  {
}

                    printf("input the age:\n");

                         scanf("%d",&p->age);

                printf("input the sex:\n");

                      getchar();

                         scanf("%c",&p->sex);

                         flag = 1;

                         break;

                    }

                    p=p->next;


               }

           if(flag==0)

               {

             printf("error username\n");

               }

      }
```

OUTPUT:



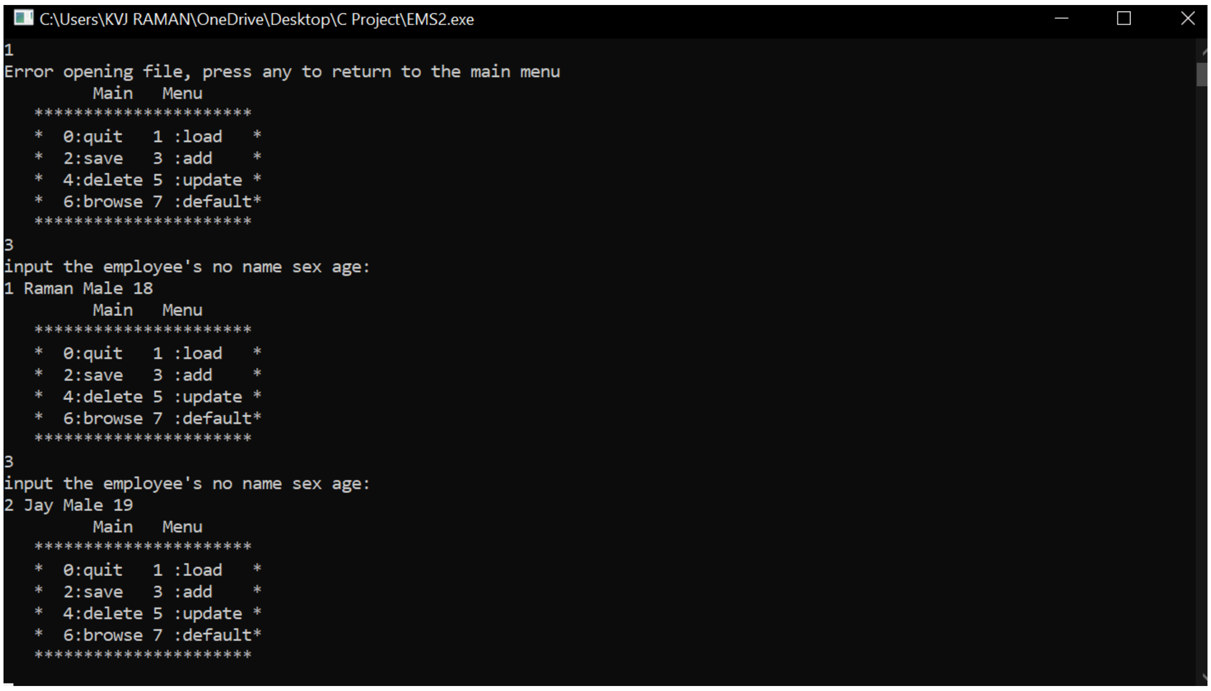
C:\Users\KVJ RAMAN\OneDrive\Desktop\C Project\EMS2.exe

```
         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
0

--------------------------------
Process exited after 11.5 seconds with return value 0
Press any key to continue . . .
```

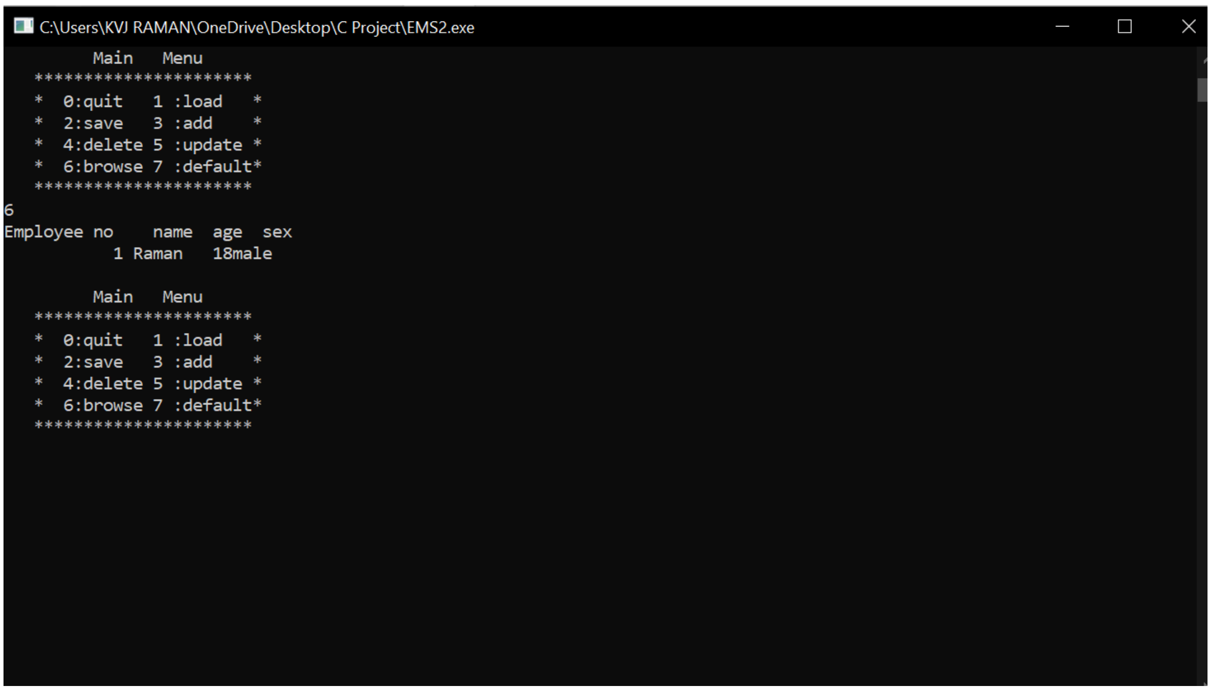


C:\Users\KVJ RAMAN\OneDrive\Desktop\C Project\EMS2.exe

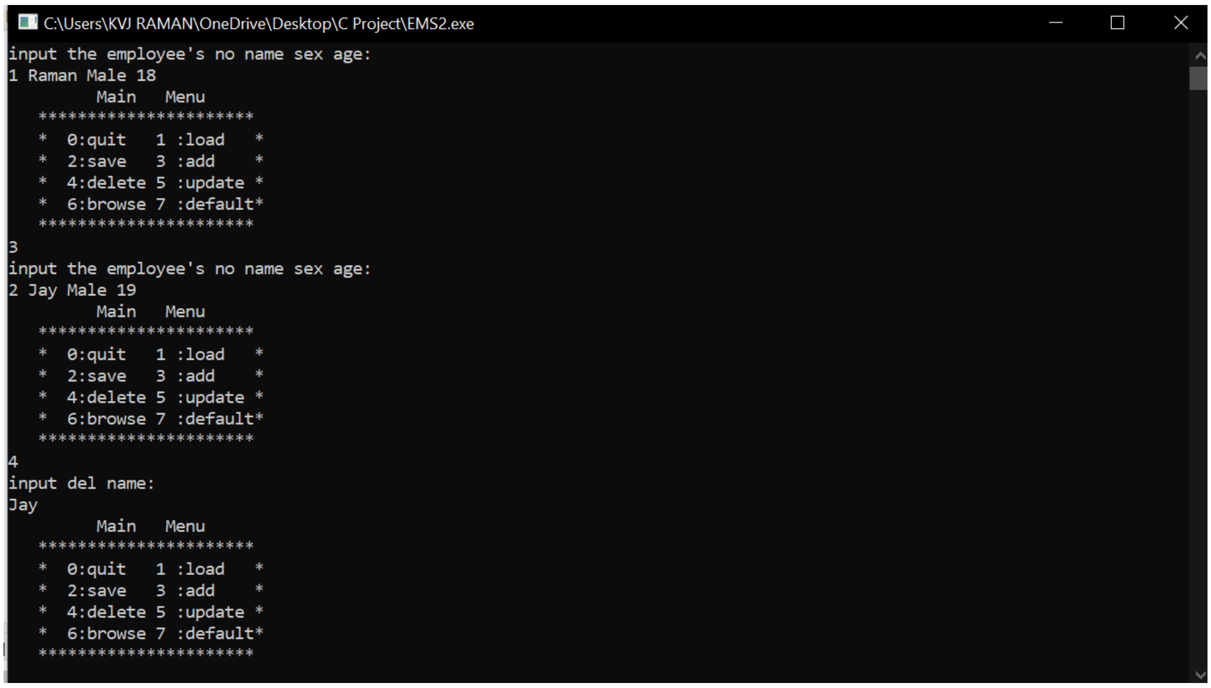
```
         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
1
Error opening file, press any to return to the main menu
```



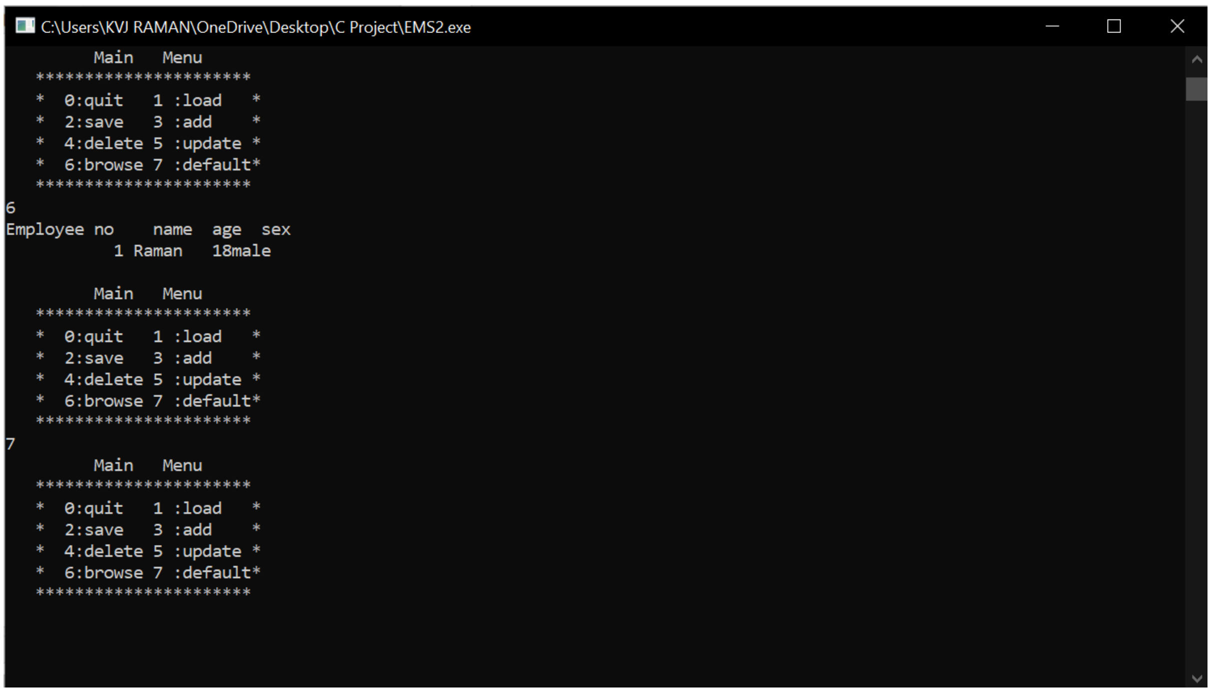
```
1
Error opening file, press any to return to the main menu
         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
3
input the employee's no name sex age:
1 Raman Male 18
         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
3
input the employee's no name sex age:
2 Jay Male 19
         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
```



```
         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
6
Employee no    name  age  sex
           1 Raman    18male

         Main   Menu
   *********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   *********************
```



```
input the employee's no name sex age:
1 Raman Male 18
         Main   Menu
   **********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   **********************
3
input the employee's no name sex age:
2 Jay Male 19
         Main   Menu
   **********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   **********************
4
input del name:
Jay
         Main   Menu
   **********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   **********************
```



```
         Main   Menu
   **********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   **********************
6
Employee no    name  age  sex
           1 Raman   18male

         Main   Menu
   **********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   **********************
7
         Main   Menu
   **********************
   *  0:quit   1 :load   *
   *  2:save   3 :add    *
   *  4:delete 5 :update *
   *  6:browse 7 :default*
   **********************
```

Conclusion

This system adopts C language traditional data structure that can be expanded by itself:Linked list implementation. Compared to using an array for storage, it is scalable! The system realizes the**Add, delete, modify, check, print out, write to disk, read disk information and other basic functions.**
The system has good scalability and can extend other functions. For example: console encryption, combined query, etc.·Multiple functions·!