1. INTRODUCTION

Employee management and payroll system is the modern computer based record management system of employees of any firm. Since it would be very difficult for any firm to maintain the records of employee on the papers and keep their attendance records also, we tried to convert this manpower to computer power. It is found that this is very efficient way to manage records as well as attendance record through this system.

1.1 Purpose / Objective:

The main goal of this project is to make the record of employee's easier & quicker.

- It is situated for all level of peoples.
- It provides proper details about the entire employee & their posts.
- User friendly environment makes the data handling more easily.
- It easily provides an environment where the user can get information about all the employees / worker

1.2 Study of existing System:

- Exiting EMS is based on the standalone system.
- It is developed on the access 95 hence it is not compatible on modern operating system.
- Existing EMS is not user friendly.
- It is not provided with the detailed project information done or to be assigned based on the application.
- It needs extra manual power also.

Scope of new System:

- It is user friendly, can be accessed by any one.
- It has user id and password system to maintain privacy and security.
- It is very fast and accurate.
- No need of extra manual effort.
- Just need little knowledge to operate the system.
- Doesn't require any extra hardware device.

1.3 Software requirements:

- Operating System : Windows 8.1
- System tool : Turbo c ++

Software Justification:

Operating System: As now a day's windows 8.1 or higher are more common in market, we have designed this software to support all these OS.

System tool: turbo c ++ is the powerful programmturing language to develop application for windows. Windows is the operating system, which runs based on messages and with rich user interface.

1.4 Hardware Requirements:

The configuration given below is the Hardware handled for the system development.

Processor	:	intel core i5
Primary Memory (RAM)	:	1 TB
Monitor	:	COLOR, 15.6"inch
Display card	:	HD DISPLAY
Mouse & Keyboard	:	Any Company

These above hardware requirements are now a days very common and can be found on any computer system. It is kept in mind while designing that no extra hardware support which can be needed to run the software should be there so that it can be afforded by any firm or industry.

ANALYSIS

The main features of this project include basic file handling operations; you will learn how to add, list, modify and delete data to/from file. The source code is relatively short, so thoroughly go through the mini project, and try to analyze how things such as functions, pointers, files, and arrays are implemented.

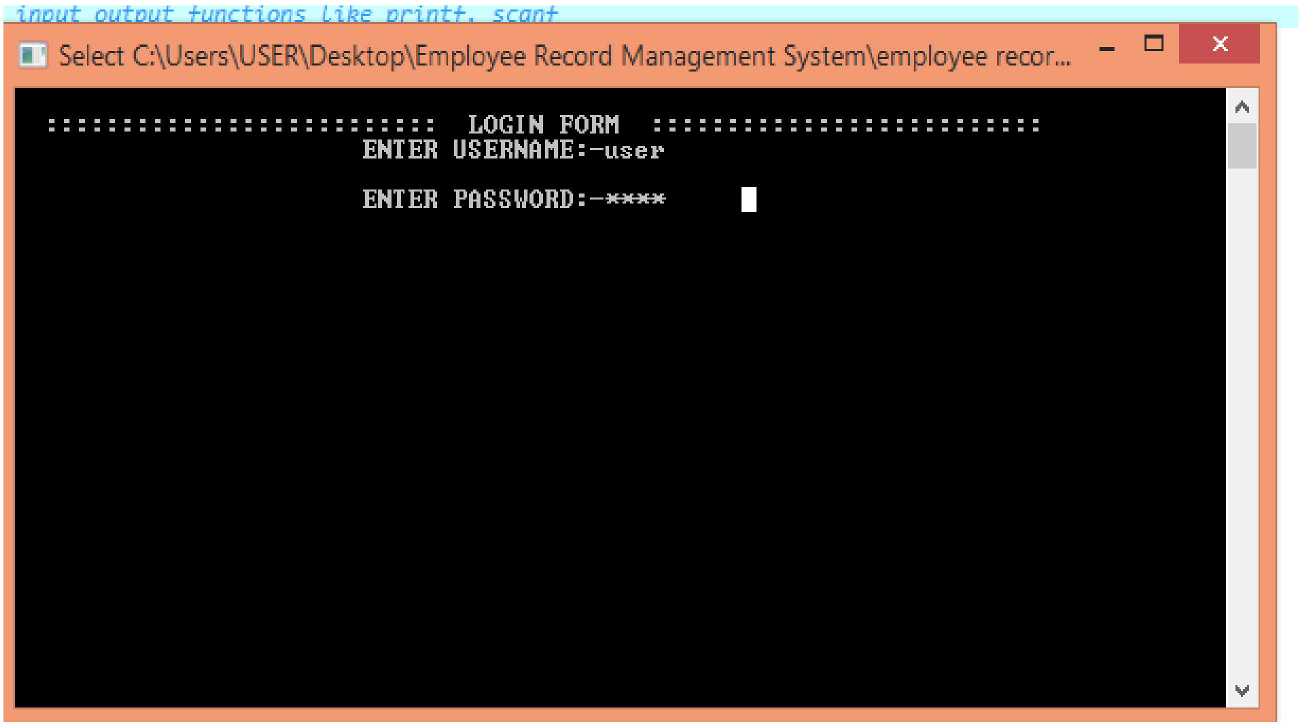
Currently, listed below are the only features that make up this project, but you can add new features as you like to make this project a better one!

- Add record
- List record
- Modify record
- Delete record

The functions used in this project are simple and they basically manipulate file handling and data structures. So, I will only describe the **gotoxy** function used in this project. Try to understand how this functions works as you may want to use it or find it used in many other C mini projects.

FRONTEND FORM SCREENS

Login Form Screen:

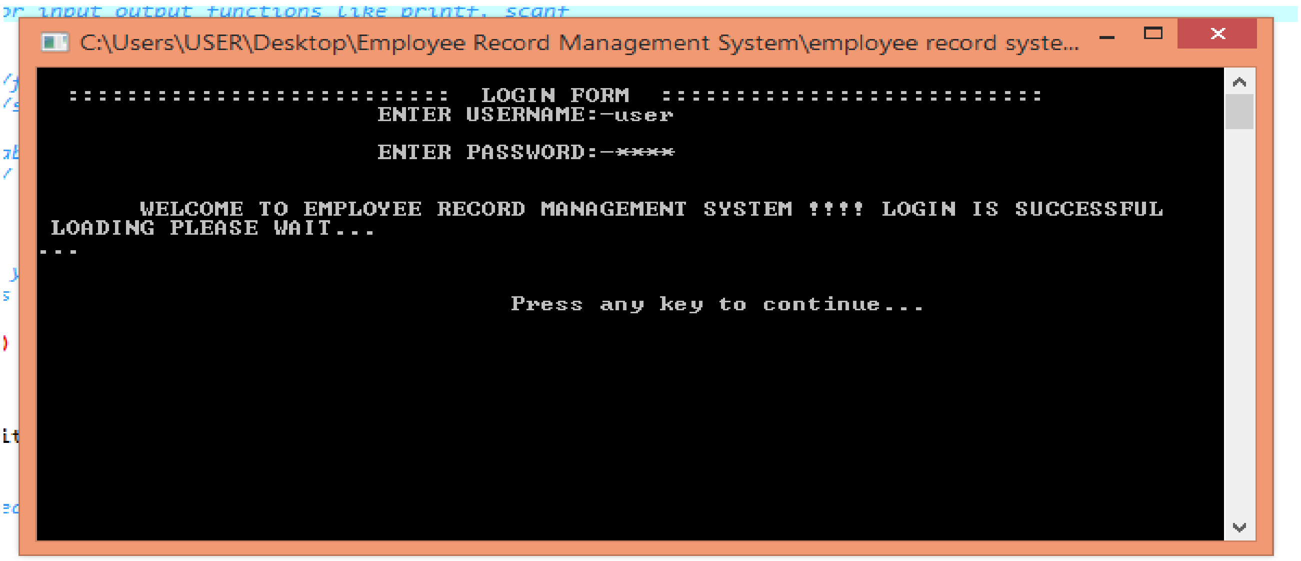


input output functions like printf, scanf

Select C:\Users\USER\Desktop\Employee Record Management System\employee recor...

```
::::::::::::::::::::::::::  LOGIN FORM  ::::::::::::::::::::::::::
                       ENTER USERNAME:-user

                       ENTER PASSWORD:-*****
```

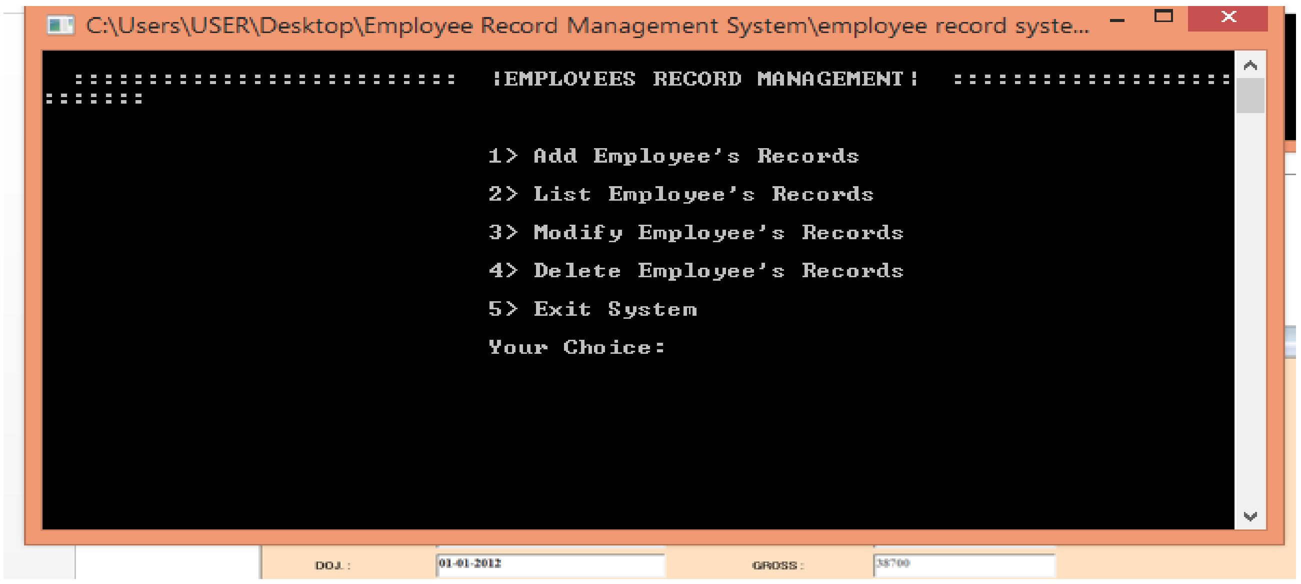
Welcome Form Screen:



C:\Users\USER\Desktop\Employee Record Management System\employee record syste...

```
 ::::::::::::::::::::::::::  LOGIN FORM  ::::::::::::::::::::::::::
                           ENTER USERNAME:-user

                           ENTER PASSWORD:-*****


      WELCOME TO EMPLOYEE RECORD MANAGEMENT SYSTEM !!!! LOGIN IS SUCCESSFUL
 LOADING PLEASE WAIT...
...

                                      Press any key to continue...
```

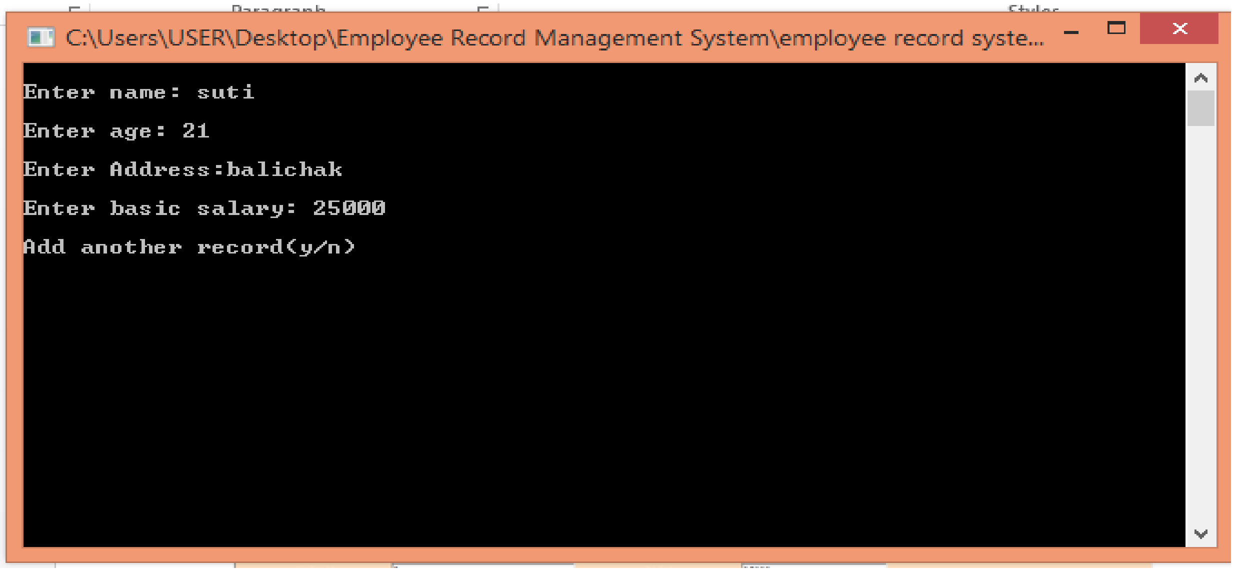
Add Employee Screen:



C:\Users\USER\Desktop\Employee Record Management System\employee record syste...

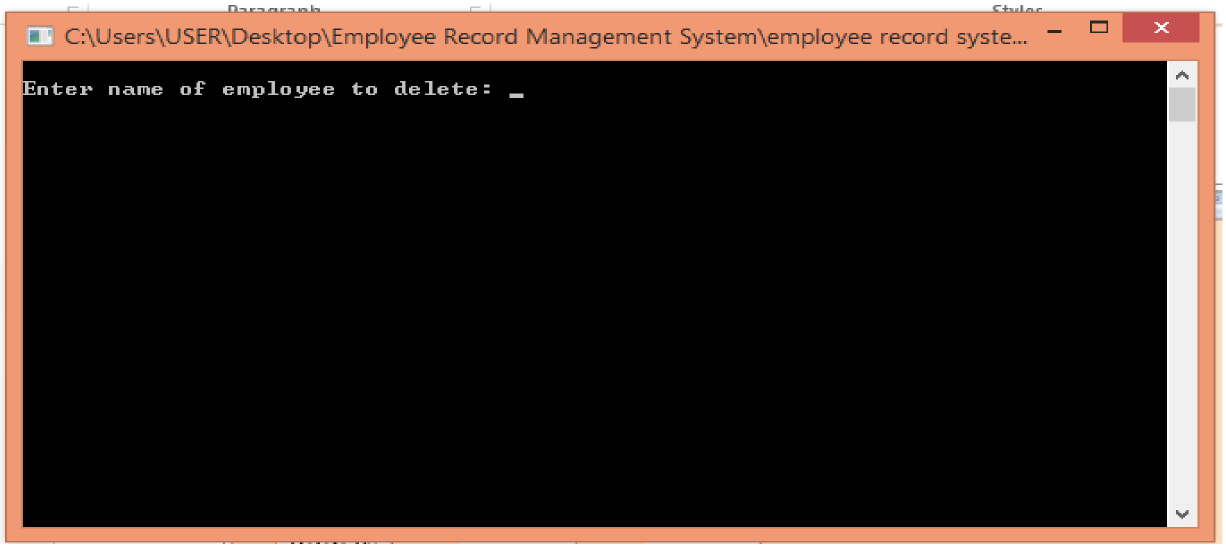
```
 ::::::::::::::::::::::::::  !EMPLOYEES RECORD MANAGEMENT!  :::::::::::::::::::
:::::::

                              1> Add Employee's Records

                              2> List Employee's Records

                              3> Modify Employee's Records

                              4> Delete Employee's Records

                              5> Exit System

                              Your Choice:
```

Modify Employee Screen:


C:\Users\USER\Desktop\Employee Record Management System\employee record syste...

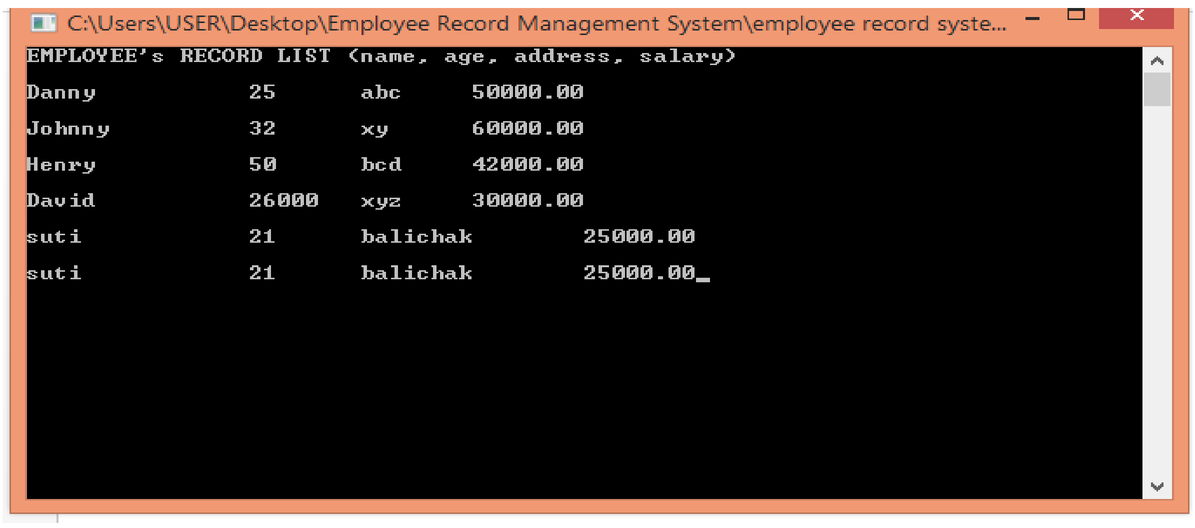
```
Enter name: suti
Enter age: 21
Enter Address:balichak
Enter basic salary: 25000
Add another record(y/n)
```

Delete Employee Screen


C:\Users\USER\Desktop\Employee Record Management System\employee record syste...

```
Enter name of employee to delete: _
```

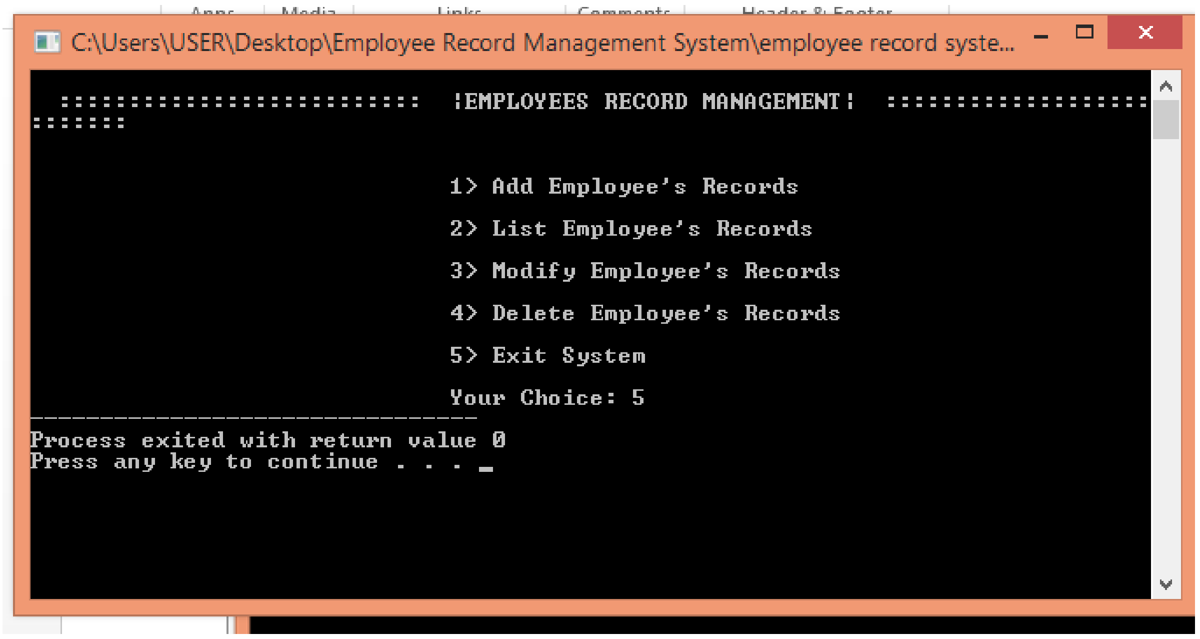
EMPLOYEE RECORD LIST



C:\Users\USER\Desktop\Employee Record Management System\employee record syste...

```
EMPLOYEE's RECORD LIST (name, age, address, salary)
Danny           25      abc     50000.00
Johnny          32      xy      60000.00
Henry           50      bcd     42000.00
David           26000   xyz     30000.00
suti            21      balichak        25000.00
suti            21      balichak        25000.00_
```

EXIST SYSTEM



C:\Users\USER\Desktop\Employee Record Management System\employee record syste...

```
  ::::::::::::::::::::::::::   |EMPLOYEES RECORD MANAGEMENT|  :::::::::::::::::::
:::::::

                              1> Add Employee's Records

                              2> List Employee's Records

                              3> Modify Employee's Records

                              4> Delete Employee's Records

                              5> Exit System

                              Your Choice: 5
--------------------------------
Process exited with return value 0
Press any key to continue . . . _
```

SOURCE CODE :

```
#include <stdio.h> ///for input output functions like printf, scanf

#include <stdlib.h>

#include <conio.h>

#include <windows.h> ///for windows related functions (not important)

#include <string.h>  ///string operations


/** List of Global Variable */

COORD coord = {0,0}; /// top-left corner of window


/**

   function : gotoxy

   @param input: x and y coordinates

   @param output: moves the cursor in specified position of console

*/

void gotoxy(int x,int y)

{

   coord.X = x;

   coord.Y = y;

   SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE),coord);

}


/** Main function started */


void login()

{

         int a=0,i=0;

   char uname[10],c=' ';
   char pword[10],code[10];

   char user[10]="user";

   char pass[10]="pass";

   do

{


   printf("\n  :::::::::::::::::::::::::  LOGIN FORM  ::::::::::::::::::::::::  ");

   printf(" \n                       ENTER USERNAME:-");

         scanf("%s", &uname);

         printf(" \n                       ENTER PASSWORD:-");

         while(i<10)

         {

            pword[i]=getch();

            c=pword[i];

            if(c==13) break;

            else printf("*");

            i++;

         }

         pword[i]='\0';

         //char code=pword;

         i=0;

         //scanf("%s",&pword);

                  if(strcmp(uname,"user")==0 && strcmp(pword,"pass")==0)

         {

         printf("  \n\n\n       WELCOME TO EMPLOYEE RECORD MANAGEMENT
SYSTEM !!!! LOGIN IS SUCCESSFUL");

            printf("\n LOADING PLEASE WAIT... \n");

   for(i=0; i<3; i++)

   {
        printf(".");
        Sleep(500);
     }
          printf("\n\n\n\t\t\t\tPress any key to continue...");
          getch();//holds the screen
          break;
          }
          else
          {
                  printf("\n        SORRY !!!!  LOGIN IS UNSUCESSFUL");
                  a++;

                  getch();//holds the screen

          }
}
          while(a<=2);
          if (a>2)
          {
                  printf("\nSorry you have entered the wrong username and password for
four times!!!");

                  getch();

                  }
                  system("cls");
}
```

```
int main()
{
        int i=0;
        login();
  FILE *fp, *ft; /// file pointers
  char another, choice;

  /** structure that represent a employee */
  struct emp
  {
    char name[40]; ///name of employee
    int age; /// age of employee
    char address[20];//address of employee
    float bs; /// basic salary of employee
  };

  struct emp e; /// structure variable creation

  char empname[40]; /// string to store name of the employee

  long int recsize; /// size of each record of employee

  /** open the file in binary read and write mode
  * if the file EMP.DAT already exists then it open that file in read write mode
  * if the file doesn't exit it simply create a new copy
  */
  fp = fopen("EMP.DAT","rb+");
  if(fp == NULL)
```

```
    {
        fp = fopen("EMP.DAT","wb+");
        if(fp == NULL)
        {
            printf("Connot open file");
            exit(1);
        }
    }

    /// sizeo of each record i.e. size of structure variable e
    recsize = sizeof(e);

    /// infinite loop continues untile the break statement encounter
    while(1)
    {

        system("cls"); ///clear the console window

        printf(" \n  ::::::::::::::::::::::::::  |EMPLOYEES RECORD MANAGEMENT|
:::::::::::::::::::::::::: \n");
        gotoxy(30,05); /// move the cursor to postion 30, 10 from top-left corner
                printf("1> Add Employee's Records"); /// option for add record
        gotoxy(30,07);
        printf("2> List Employee's Records"); /// option for showing existing record
        gotoxy(30,9);
        printf("3> Modify Employee's Records"); /// option for editing record
        gotoxy(30,11);
        printf("4> Delete Employee's Records"); /// option for deleting record
        gotoxy(30,13);
```

```
printf("5> Exit System"); /// exit from the program

gotoxy(30,15);

printf("Your Choice: "); /// enter the choice 1, 2, 3, 4, 5

fflush(stdin); /// flush the input buffer

choice  = getche(); /// get the input from keyboard

switch(choice)



{

case '1':  /// if user press 1

   system("cls");

   fseek(fp,0,SEEK_END); /// search the file and move cursor to end of the file

   /// here 0 indicates moving 0 distance from the end of the file


   another = 'y';

   while(another == 'y')  /// if user want to add another record

   {

      printf("\nEnter name: ");

      scanf("%s",e.name);

      printf("\nEnter age: ");

      scanf("%d", &e.age);

      printf("\nEnter Address:");

      scanf("%s",e.address);

      printf("\nEnter basic salary: ");

      scanf("%f", &e.bs);


      fwrite(&e,recsize,1,fp); /// write the record in the file
```

```
            printf("\nAdd another record(y/n) ");
            fflush(stdin);
            another = getche();
           }
           break;
        case '2':
           system("cls");
           printf("EMPLOYEE's RECORD LIST (name, age, address,
salary)");
           rewind(fp); ///this moves file cursor to start of the file
           while(fread(&e,recsize,1,fp)==1)  /// read the file and fetch the
record one record per fetch
           {


            printf("\n\n%s \t\t%d \t%s \t%.2f",e.name,e.age,e.address,e.bs); /// print the
name, age and basic salary
          }
          getch();
          break;


        case '3':  /// if user press 3 then do editing existing record
          system("cls");
          another = 'y';
          while(another == 'y')
          {
            printf("Enter the employee name to modify: ");
            scanf("%s", empname);
            rewind(fp);
            while(fread(&e,recsize,1,fp)==1)  /// fetch all record from file
            {
              if(strcmp(e.name,empname) == 0)  ///if entered name matches with that in
file
              {
                printf("\nEnter new name,age,address and bs: ");
                scanf("%s%d%s%f",e.name,&e.age,&e.address,&e.bs);
                fseek(fp,-recsize,SEEK_CUR); /// move the cursor 1 step back from
current position
                fwrite(&e,recsize,1,fp); /// override the record
                break;
              }
            }
            printf("\nModify another record(y/n)");
            fflush(stdin);
            another = getche();
          }
          break;
        case '4':
          system("cls");
          another = 'y';
          while(another == 'y')
          {
            printf("\nEnter name of employee to delete: ");
            scanf("%s",empname);
            ft = fopen("Temp.dat","wb");  /// create a intermediate file for temporary
storage
            rewind(fp); /// move record to starting of file
            while(fread(&e,recsize,1,fp) == 1)  /// read all records from file
            {
              if(strcmp(e.name,empname) != 0)  /// if the entered record match
```

```
                {
                    fwrite(&e,recsize,1,ft); /// move all records except the one that is to be
deleted to temp file
                }
            }
            fclose(fp);
            fclose(ft);
            remove("EMP.DAT"); /// remove the orginal file
            rename("Temp.dat","EMP.DAT"); /// rename the temp file to original file
name
            fp = fopen("EMP.DAT", "rb+");
            printf("Delete another record(y/n)");
            fflush(stdin);
            another = getche();
          }
          break;
        case '5':
          fclose(fp);  /// close the file
          exit(0); /// exit from the program
        }
    }
    return 0;
}
```

LIMITATIONS AND FURTHER ENHANCEMENTS

Limitations:

- Skyworld Tech Employee Management System is limited to our control. Any change or modification can be done by us only.
- It is not an online application or system.
- We have not provided any backup system for this.
- The system need to run oracle on background.

Further Enhancements:

- Search details can be enhanced by detailed identification of searching problems and rectification steps by company.
- This system is entirely designed for STAND ALONE usage, according to the need of the customer if needed the same system can be designed with Network facility to handle the branch activities also.
- To make this system Online and creating data backups.
- Integrated Password management facility can be created.

CONCLUSION

The Employee Management System didn’t automate 100% of their work, but it is really a good start to computerize everything and entire Detail can be 100% computerized.

As far as the work done so far much care was given about the user friendliness and a very good interaction with the end users. The interface are so designed and channeled the users can never make any mistake while using the application, for an example while adding new record, user can’t go out without either saving or canceling the operation, till the time either they save or cancel the current operation all other operations are blocked. There is no chance of making any mistake in the application.

Bibliography & References

- LET US C - Yashvant Kanitkar
  (A Book of Basic C lang)
- www.codewithc.com