EMPLOYEE RECORD SYSTEM REPORT

BY

Isai Tillothama

INTRODUCTION

Employee Record System is software built to handle the primary housekeeping functions of a company. ERS helps companies keep track of all the employees and their records. It is used to manage the company using a computerized system. This software built to handle the records of employees of any company. It will help companies to keep track of all the employees’ records in a file.

Aim of the Employee’s Record System: The user will be provided with 5 options:

Add a new record.

Delete a record.

Modify a record.

View all the records.

Exit.

Data of the Employees:

Name.

Age.

Salary.

Employee ID.

ABSTRACT

his project report provides a detail of what I found out about the employee records management system that exists at UCU and provides a theoretic layout of the electronic system that I designed at the end of the project. The content of this report are summarized in various chapters;

* provides the introduction of the topic that is the employees’ record management system and the background of the case study, Uganda Christian University.
* seeks to give the various authors’ study about the same topic.
* states the research design and methodology that was used in my study.
* clearly explains the system analysis and design 5 shows how the system design was implemented and the screen shots of the system’s major functionalities.

contains the conclusions and recommendations about the same project

* In the appendices included area samples of research instrument, the requisition letter for research.

#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 \*/

int main()

{

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

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

gotoxy(30,10); /// move the cursor to postion 30, 10 from top-left corner

printf("1. Add Record"); /// option for add record

gotoxy(30,12);

printf("2. List Records"); /// option for showing existing record

gotoxy(30,14);

printf("3. Modify Records"); /// option for editing record

gotoxy(30,16);

printf("4. Delete Records"); /// option for deleting record

gotoxy(30,18);

printf("5. Exit"); /// exit from the program

gotoxy(30,20);

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 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");

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%s %d %.2f",e.name,e.age,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 and bs: ");

scanf("%s%d%f",e.name,&e.age,&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

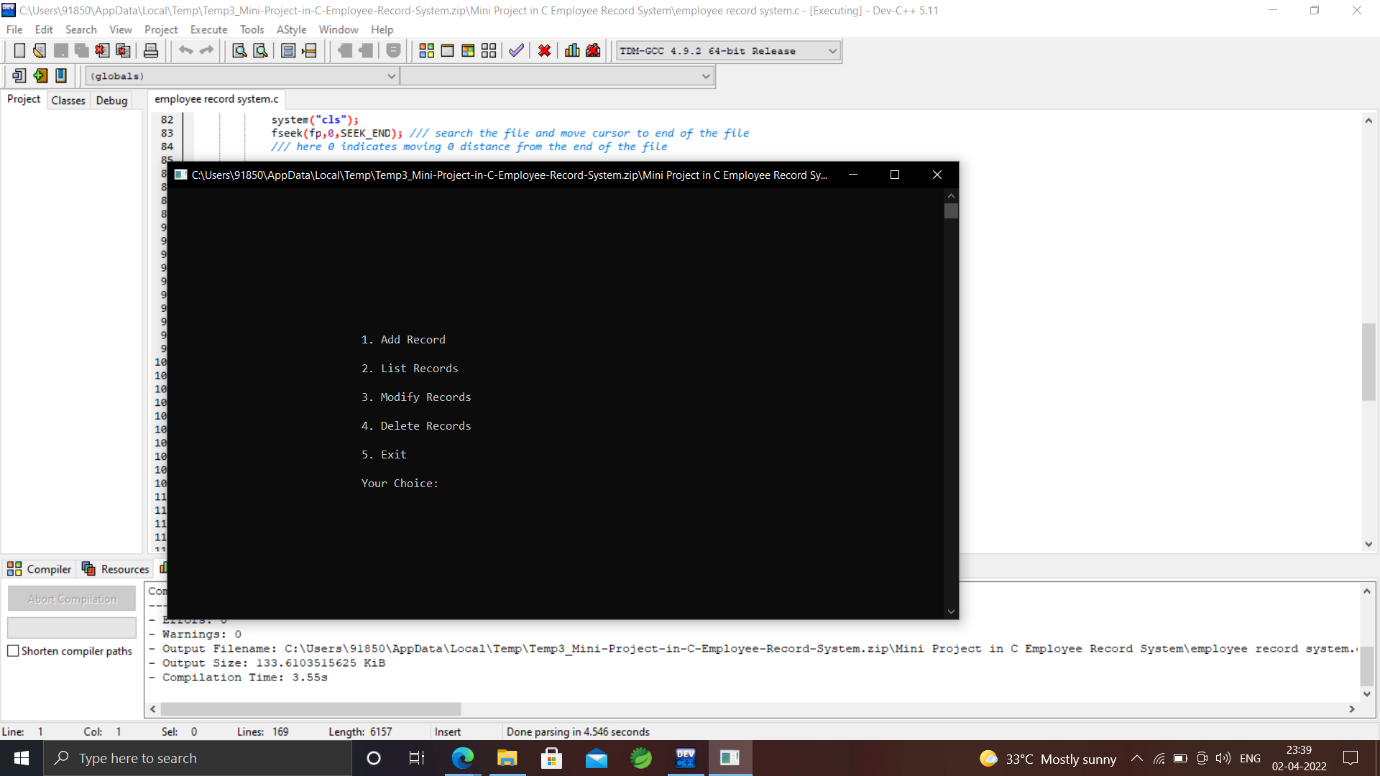
}

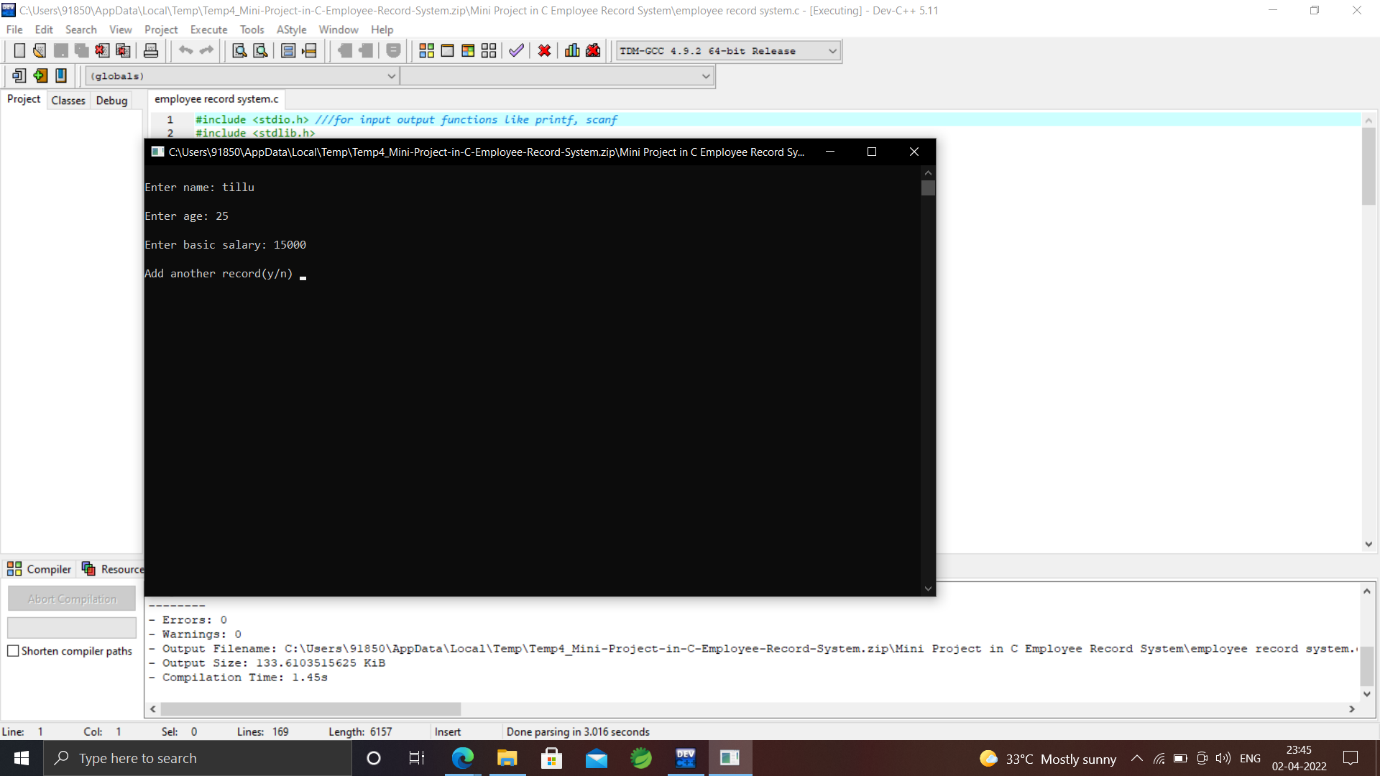
}

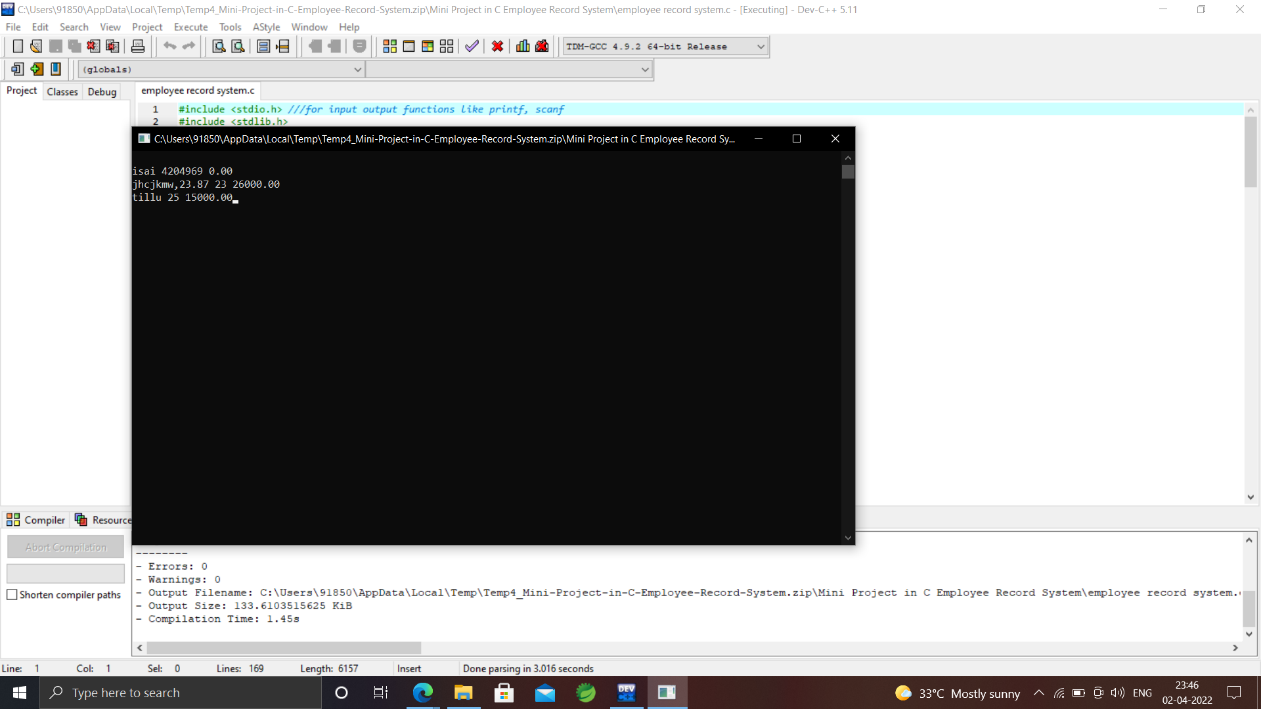
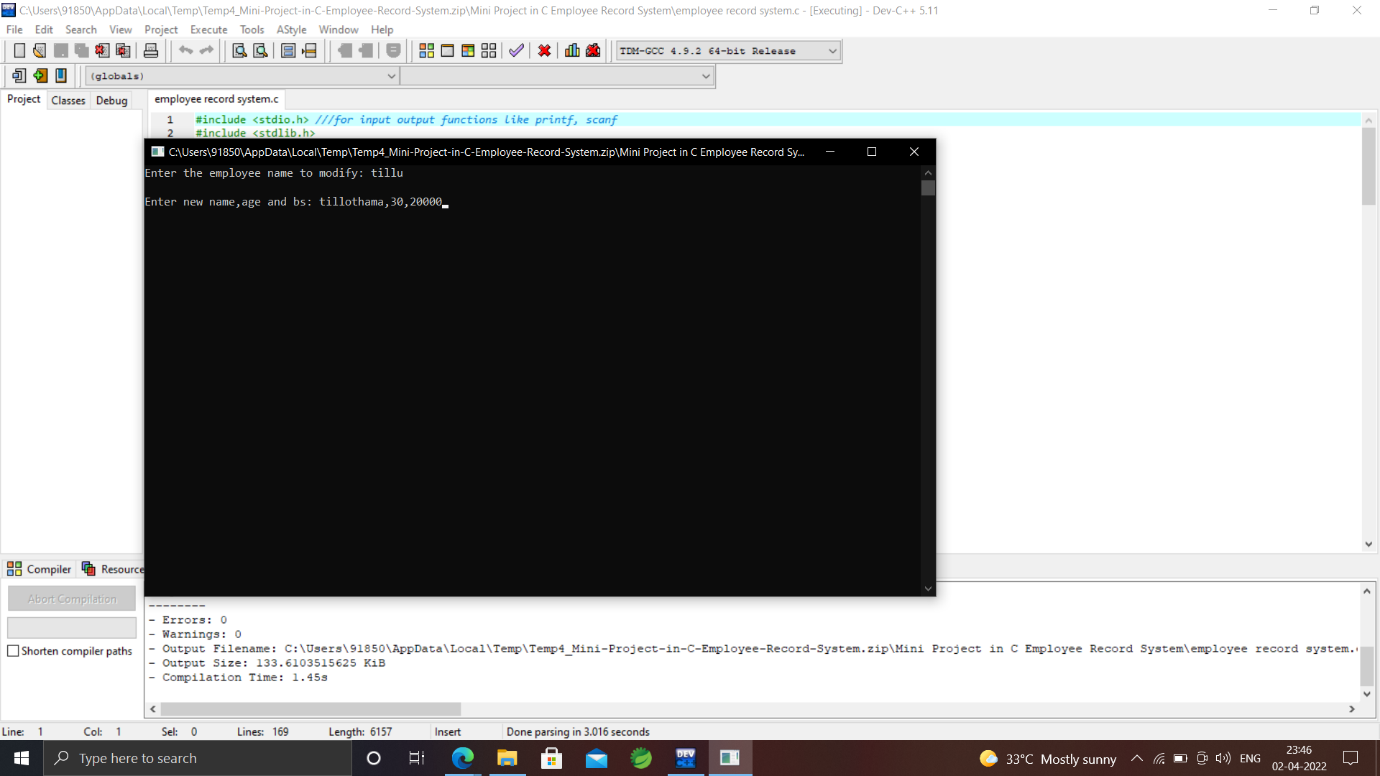
return 0;

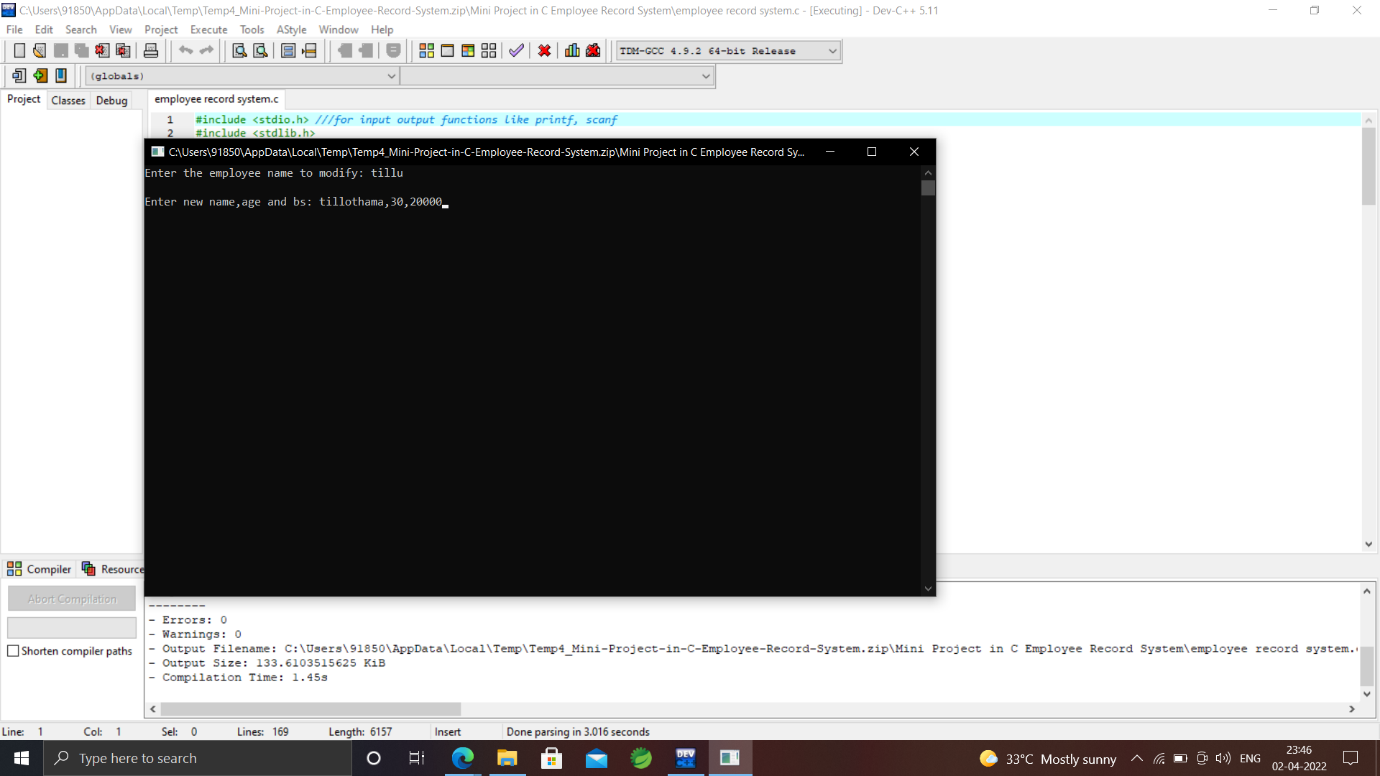
}

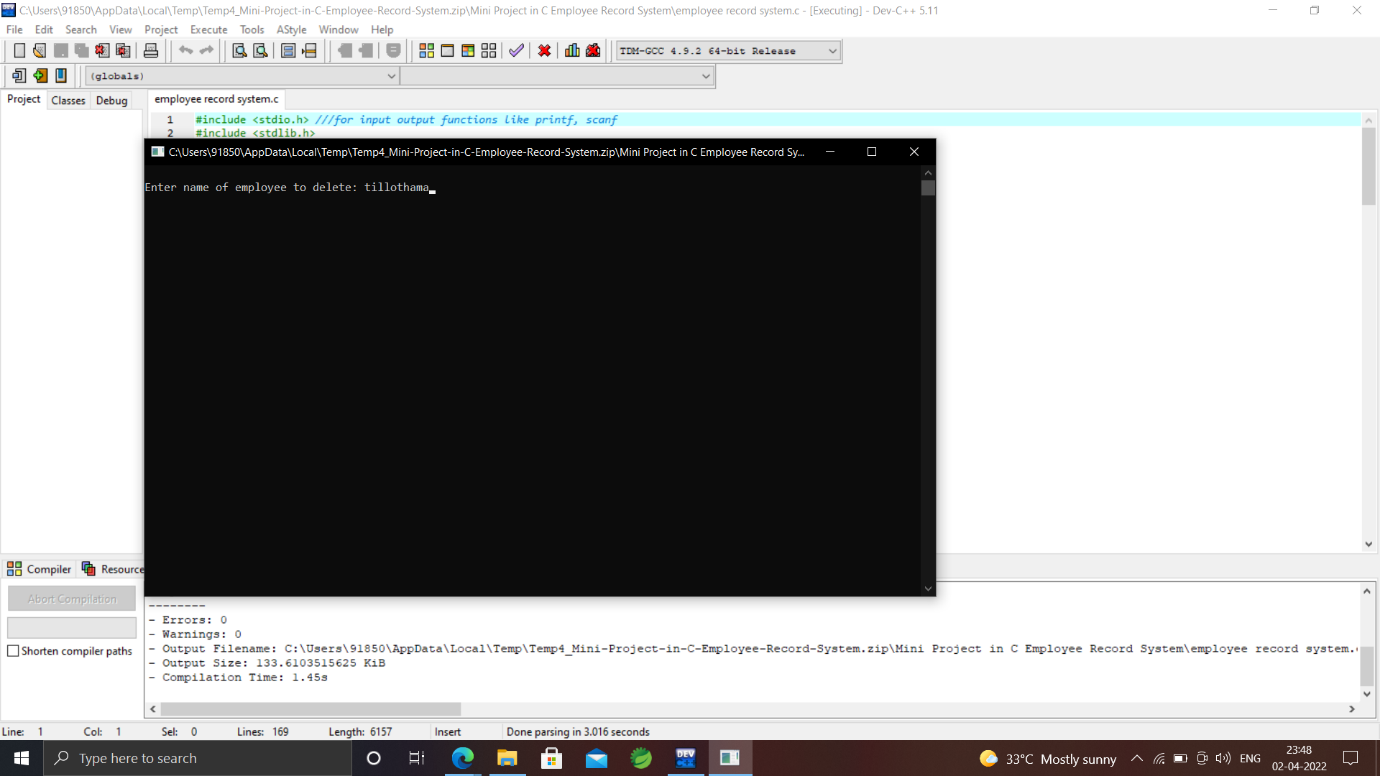
OUTPUT SCREENSHOTS

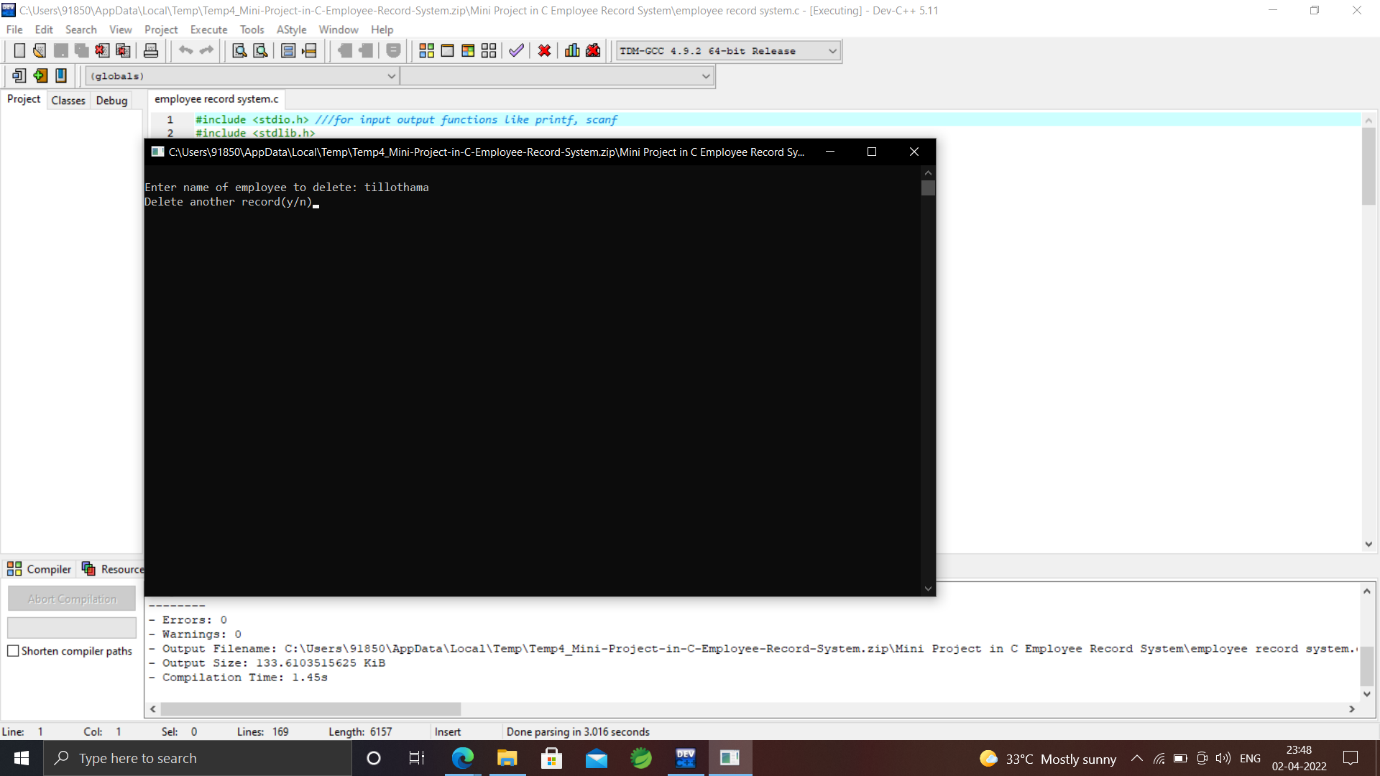


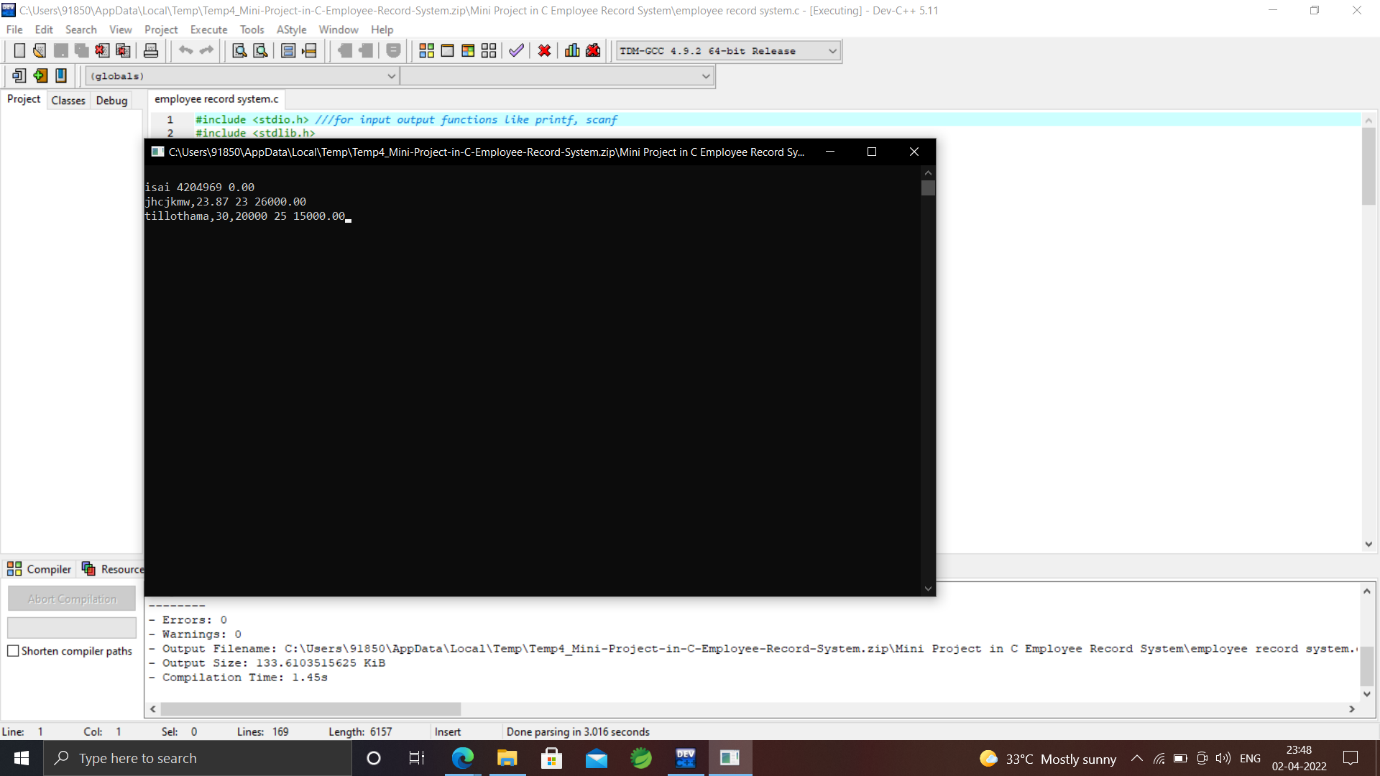


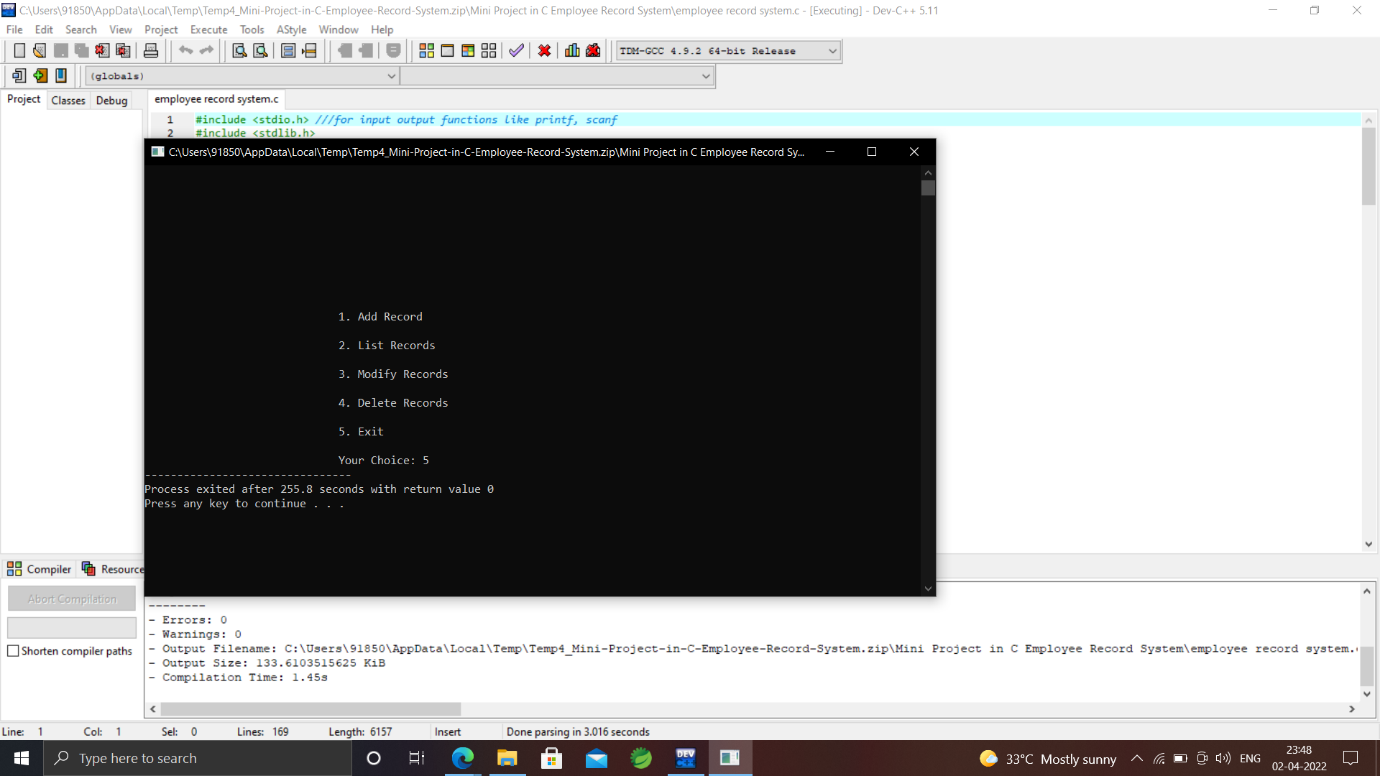


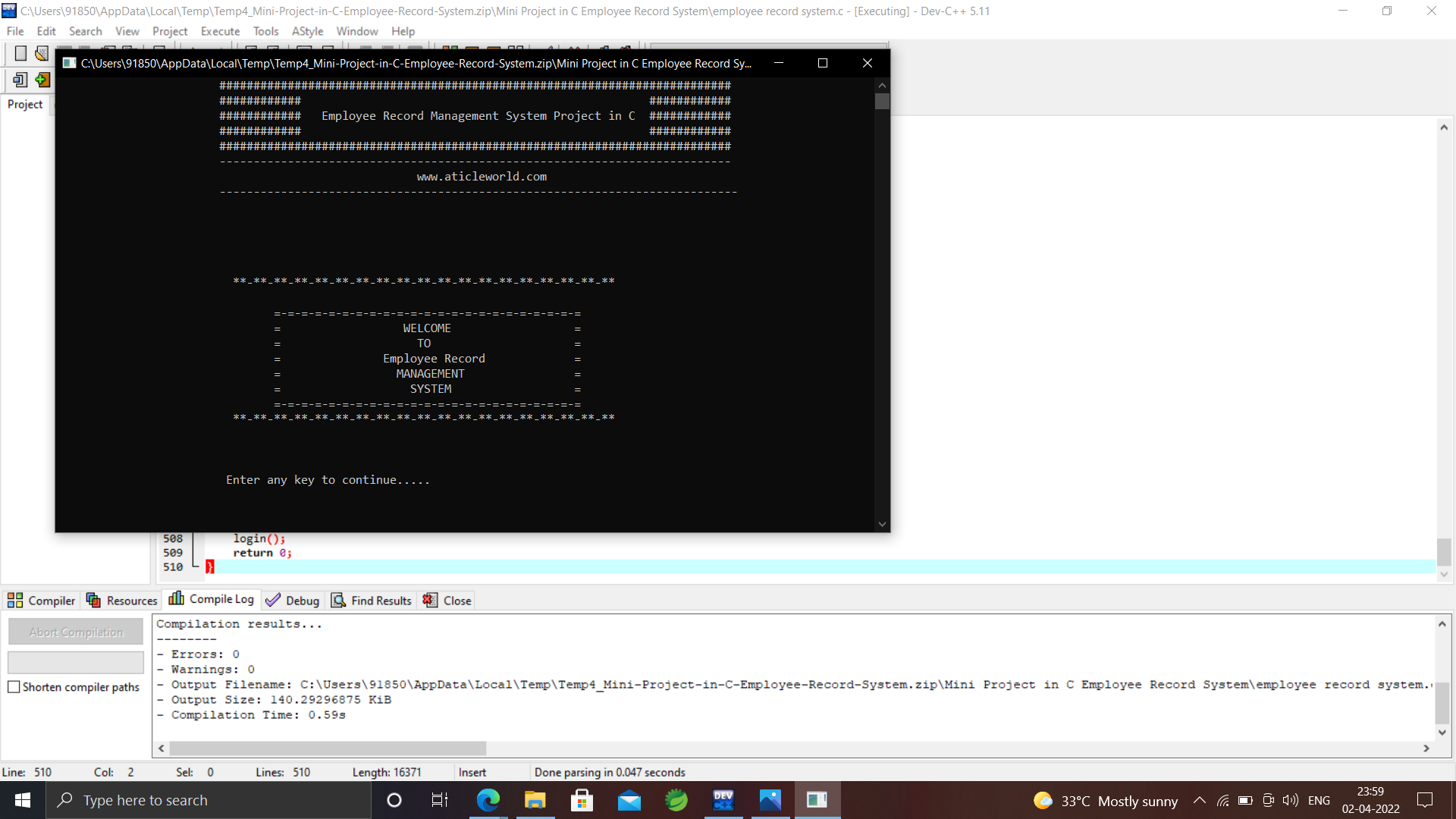












CONCLUSION

There were so many challenges faced in the existing system that needed to be addressed and these have given rise to development of this system. The challenges helped determine what the developed system addresses. In case of new challenges sighted with use of the new system, this calls for system upgrade or addition or more features to mitigate such problems.

The employee Records management system developed is a secure and user friendly application which can be used to manage employee records effectively and efficiently.

The developed system is subject to development by integration of more other special features and so it’s open for more study.

The developed system is flexible and can therefore be customized for use by other organizations to manage employee records effectively.

Conclusively, proper utilization of this system will help optimize resources at the organization and increase the effective flow of business since human resource management lies at the fore front of organizational growth and this is well achieved by having an efficient mechanism of managing their records.