# A MINI PROJECT REPORT

**ON**

# Attendance System Using File Handling In C

Submitted to Mumbai University

In the partial fulfillment of the requirement for the award of the degree of

**Bachelor of Engineering**

In

**COMPUTER ENGINEERING**

By

**Mr. Shaikh Shadab Ali**

**(17DCO74)**

Under the guidance of

**Mr. Muhammed Salman Shamsi**

**Assistant Professor**

--

## Department of Computer Engineering

## Anjuman-I-Islam Kalsekar Technical Campus

## Affiliated to Mumbai University

KHANDA GOAN, NEW PANVEL, NAVI MUMBAI, MAHARASHTRA

2017-2018

# Department of Computer Engineering

# Anjuman-I-Islam Kalsekar Technical Campus

# Affiliated to Mumbai University

KHANDA GOAN, NEW PANVEL, NAVI MUMBAI, MAHARASHTRA

2017-2018


## DECLARATION BY THE CANDIDATE

Shadab Ali bearing Roll number: 17DCO74, hereby declare that the mini project report entitled **"Attendance System Using File Handling In C"**, is a record of bonafide work carried out by me and the results embodied in this project have not been reproduced or copied from any source. The results of this project report have not been submitted to any other University or Institute for the award of any other Degree or Diploma.

**Shadab Ali**

**(17DCO74)**

# Department of Computer Engineering

# Anjuman-I-Islam Kalsekar Technical Campus

# Affiliated to Mumbai University

KHANDA GOAN, NEW PANVEL, NAVI MUMBAI, MAHARASHTRA

2017-2018

## CERTIFICATE

This is to certify that the project report entitled **"Attendance System Using File Handling In C"**, submitted by **Mr.Shadab Ali**, bearing **Roll. No.: 17DCO74** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Computer Engineering** is a record of bonafide work carried out by him.

**Course Owner**

**(Assit. Prof. Muhammed Salman Shamsi)**

# INDEX

## CONTENTS

# Chapter 1

# INTRODUCTION

## 1.1 Introduction

Every colleges have variety of work to do. Like maintaining database of students, arranging documents, organizing different-different events. For this purpose, our project comes into picture. Maintaining database includes many operations in which one operation is to take attendance of student. So, we decided to build a project in c using file structure which will meet the need of this college, with best suitable GUI (graphical user interface), and also some enriching features which will ease the redundant works and increase the reliability of taking attendance of student.

As far as implementation is concerned, we have opted to take C language and mainly used two types of structure viz, Arrays and FILE. We have also emphasized on various real-time problem to be tackled while building this project such as permanent access of data, authorization and authentication, and also all the CRUD operation i.e. (Creation, Deletion, Updating, Reading).

The key feature in our project a user (in our case its respected faculties), will get a permanent file as a record of the attendance with current date and time inside that particular file and the file name being saved as the current date and time (in minutes) as a text file. Deleting and reading can be done on those file from this project/software itself. Also, separate login module is implemented to ensure security and only respected records can be viewed by authenticated and authorized personalities.

## 1.2 Scope

Features of Our Project

- Fully authorized and authenticated system
- Permanent access of files to only respective authorized persons within the software/project itself
- ALL CRUD operation is implemented on the records.
- Separate database for student, increasing the dynamicity of software
- Time to time user friendly guidelines and messages so any individuals can easily handle our system without any difficulties
- Records being saved with current date and time as filename, and also inside record with respective faculty name and their subject code for future security purposes
- No need of executing the project or running the project from a specified directory. It can be run from any directory and any system so well implemented deployment principles

Our project requirement included variety of modules and functionalities to meet such as creation of files, updating of created records, deleting records, reading of files. All these operations should be done by respected faculties after login. The student details such as roll number and name should be prompted from student database which should be editable database as student database will vary from time to time year to year.

As a result, we have successfully overcome all these barriers and implemented a fully functional software. This software can easily be used by all the respected faculties whose separate login is created regardless of the fact the respective persons/user knows to use a computer system or not with very well-suited user-friendly messages and guidelines throughout the flow of the project.

There are some already developed project in C++ or C but the problem in those is everything is hardcoded, and no dynamicity is achieved on those projects. This project is created keeping in mind with for a particular class, but it can be expanded for two classes, or two branches or for full college campus as all the details of student will be generated from the student database and then accordingly attendance will be taken with proper class name and date and time.

## 1.3 Problem Statement

Taking attendance is the process or act to make a record of who is present or absent. Example a faculty takes attendance of students regularly.

This project requires us to develop an application, that will take the attendance of student with the help of computer system and will save the records electronically, securely with full authentication and authorized manner.

Following are the key points that this software includes or demand.

- The end user will be respected faculties, the faculties may be not much aware to computer system, so the system should be as user friendly as possible encouraging more and more users to use the system efficiently using this application instead of traditional mechanism of taking attendance of student in a piece of paper, without any hesitation and should be easy to handle
- Separate login mechanism for each user, to implement full authorization and authentication system
- It may be possible while inserting records or while taking records the user mistakenly puts wrong status of the students viz absent and/or present, so this problem must be avoided should be handled properly
- As many users are more used to the traditional method of taking attendance of students, the system should take care of all the drawbacks like
    - System should provide permanent access of the record electronically
    - Quick access of the particular record to respective users at required time
    - The system must be reliable enough to keep all the essential details of record like student database, faculty database, student's attendance records and all the system must implement the principle of confidentiality viz. only the intended recipients must be able to view his/her student's records.
- The system shall also be compact and should not ask for large number of resources, should be portable enough and can be run from any machines provided with some pre- requisites

# Chapter 2

# SYSTEM SPECIFICATION

## 2.1 System Requirements

### Hardware Requirements

- Intel Pentium 4 processor
- 512 MB disk space
- 256 MB RAM
- Normal PS/2 or any USB keyboard

### Software Requirements

- Microsoft Windows XP or Ubuntu 14.0 LTS
- Any utility like Notepad to view the text file
- C Compiler such as Turbo C or any IDE such as geany
- Time.h, dirent.h, string.h library for generation of directory, time and string comparison and concatenation operation.

# Chapter 3

# SYSTEM IMPLEMENTATION

## 3.1 Modules In The System

In our project there are various key modules such as separate login module for respective faculties. Validating login details, and accordingly giving access to authorized users (Faculties). After accessing the system, a user will have access to four main modules such as reading of records, creating a new record, deletion of a record and logging out.

- **Authentication Module**

  The first thing that a user will see from this application is to enter his/her valid credentials. If mistakenly a user enters valid id but wrong pass or pass that is not assign to that particular user a message will appear invalid details implementing both authentication and authorization. After successful login, a user will observe four main menus viz, creation of new record, deletion of existing record, reading of existing record and logout.

  When selecting logout, a user will be disconnected from the system. In this case, our user total is 5 and there are 5 subjects, and accordingly modules are separated accessing the database and creating a permanent file in respective user's database.

- **Reading Module**

  When selecting reading of records, respected faculties will be able to access his/her database(directory) and all the files present in that database will be displayed, then a user will have to insert the required date and time to view the records.

- **Creation Module**

  When selecting creating a new record, respected faculties will be able to create a new record, all the details will be fetched from the student database and displayed to user line by line and according to user action present and absent will be inserted after the user info. As soon as the procedure is completed the system will ask to commit (make permanent file) of this record or to update this record for any mistake occurred. Before the data is fetched from the student database, system will ask for subject code and if that subject code matches the user login, then only the system flow will continue. While the new file is created there will be additional stuffs also inserted into that particular file. On top it will be subject code that user enters and his/her name being fetched from the system itself

comparing with subject code.at bottom there will be class name, and at which time the attendance was taken with proper date, time and second in 12 hour format.

- **Updating Module**

  When a user commits a record, a permanent file will be available with the current date and time as file name.

- **Deletion Module**

  When selecting deleting of a record, respected faculties will be able to delete the existing record present in his/her database. The user has to enter the date and time to access that record and after accessing the system will again “is the user sure to delete that file the recovery process isn’t possible”.

- **Validation Module**

  Throughout this flow of project whenever willingly or unwillingly a user enters a key that is not required or not ask they will get a user-friendly message

  While entering present and absent of student, only two keys will work rest all will be inactive. One key is for present and one key is for absent.

## 3.2 CODE

```
#include<stdio.h>                    //using also to remove files
#include<string.h>                   //using to compare strings and concating
#include<time.h>                     //using to generate time
#include<conio.h>                    //using to make the screen hold
#include<dirent.h>                   //using for directory
#define esc 27
#define EOL '\n'
int main()
{
        int x=00,y=00;               //initializing variables for counter
        char id_ag[]="14FCCOAG";          //faculties id
        char id_jks[]="14FCCOJKS";
        char id_mrk[]="14FCCOMRK";
        char id_sa[]="14FCCOSA";
        char id_tk[]="14FCCOTK";
        char pw_ag[]="ag";                //faculties pass
        char pw_jks[]="jks";
        char pw_mrk[]="mrk";
        char pw_sa[]="sa";
        char pw_tk[]="tk";
        char subjcode1[]="CLS301";                //subject code
        char subjcode2[]="CLS302";
        char subjcode3[]="CLS303";
        char subjcode4[]="CLS304";
        char subjcode5[]="CLS305";
        char subjc1[100];            //taking subject code from user
        char temp_id[9];             //accepting id and pass from users
        char temp_pw[10];
        char chx;
        x1:printf("Enter  Your Credentials\n");
        printf("Login ID =>");
```

```
scanf("%s",temp_id);
printf("Password =>");
scanf("%s",temp_pw);
char ch,ch2,ch3,ch4,ch5,ch6,ch7,ch8,l1;
//assigning the character value to these variables that getch() is returning
FILE *fp2;                                  //Creating a file structure pointer
int a=(strcmp(id_ag,temp_id));
//Comparing two strings id
if(a==0)
        //if returning true
{
        int p1=(strcmp(pw_ag,temp_pw));
//comparing two strings password
        if(p1==0)
        //if returning true
        {
                printf("\nWelcome Apeksha Madem!!!\n");
                label2:printf("\nPress q to search for records\nPress n to create
                new record\nPress z to delete any records\nPress x to Logout\n");
                label1:ch2=getch();
                if (ch2=='q')           //if user enters q i.e. for reading
                do{
                {
                        DIR *d;
                        struct dirent *dir;
                        d=opendir("..//DSUMini/Faculty/Apeksha/");
                        //opening directory of speicified path
                        if(d)
                        {
                                printf("\nTotal records In ur Database are as
                                follows\n");
                                while((dir=readdir(d))!=NULL)
                                {
```

```
                printf("%s\n", dir->d_name);
        //displaying each files present in that directory
        }
        closedir(d);
}
char filename[50];
char c1[]="..//DSUMini/Faculty/Apeksha/";
//path of the faculty database
printf("\nEnter the date and time in the following format DD-MM-YYYY-HH-AM/PM\nExample: 21-08-2017-08-AM To open any specific records\n");
scanf("%s",filename);
//time and date
strcat(c1,filename);
//concat path and time and date and storing in c1
char c2[]=".txt";
strcat(c1,c2);
//concat c1 with .txt and storing in c1
fp2=fopen(c1,"r");
        //opening file in read mode
        if(fp2==NULL)
        {
        printf("\nNo such records found plz try again\npress b to go back or any key to contiue\n");
                //if no records found
        ch7=getch();
        if(ch7=='b')
        {
        goto label2;
        //when pressed b going back to main menu
        }
        }
else
{
```

```
                        ch2=fgetc(fp2);
                        //when file found and reading full files
                        while(ch2!=EOF)
                        {
                                putchar(ch2);
                                //same as printf function
                                ch2=fgetc(fp2);
                        }
                        fclose(fp2);
label3:printf("\nPress b to go back to main menu or press e to Logout\n");
                        ch3=getch();
                        if(ch3=='b')
                        {
                                goto label2;
                                //going back to main menu when pressed b
                        }
                        else if(ch3=='e')
                        {
                                goto x1;
//if pressed e then logout going back to login credential phase
                        }
                        else
                        {
                        printf("\nPlease select only above mentioned option\n");
                                goto label3;     //if any unwanted key pressed again
                repeating procedure from asking user to press b or e
                        }
                        }
                }
        }while(fp2==NULL);
                else if(ch2=='n')
                {
                //if user is creating a new record
                label81:printf("\nEnter the COURSECODE (ALL In Caps)\n");
```

```
scanf("%s",subjc1);                    //accepting coursecode
int c=(strcmp(subjcode1,subjc1));      //comparing user course code with valid course code
if(c==0)
//if returns true
{
time_t raw;
labelitr:time(&raw);
struct tm *time_ptr;
time_ptr=localtime(&raw);//declaring and using structure of time
char date[50];
strftime(date,50,"%d-%m-%Y-%I-%M-%S-%p",time_ptr);//formatting the time and storing in an array with cuurent date,month,year,hour(24 hour format),minutes,seconds,am/pm
char date11[50];
strftime(date11,50,"%d-%m-%Y-%I-%p",time_ptr);//formatting the time and storing in an array with cuurent date,month,year,hour(24 hour format),am/pm
char date1[50];
strftime(date1,50,"%d-%m-%Y",time_ptr);    //formatting the time and storing in an array with cuurent date,month,year
char time1[50];
strftime(time1,50,"%I:%M:%p",time_ptr);    //formatting the time and storing in an array with cuurent hour(24 hour format),minutes,seconds,am/pm
char d1[]="..//DSUMini/Faculty/Apeksha/temp";
//path of temporary directory
char c1[]="..//DSUMini/Faculty/Apeksha/temp/";
//path of temporary directory to access that file
char c11[]="..//DSUMini/Faculty/Apeksha/";
//path of database
strcat(c1,date);
//concatinating temporary dir path with the first strftime
```

```
        strcat(c11,date11);
                //concat permanent database path with 2nd strftime
        char c2[]=".txt";
        char c22[]=".txt";
        strcat(c1,c2);
        strcat(c11,c22);
//concat with the above mentioned path with .txt to get complete .txt file
        char p;
        char present[]="\tPresent";
        char absent[]="\tAbsent";
        //declaring string with empty array for present and absent
        FILE *fp1;
        fp1=fopen("..//DSUMini/Stud_Db.txt","r");
//creating a file ptr and opening the specified path file in read mode
        mkdir(d1);
//making a temp directory
        FILE *fp;
        fp=fopen(c1,"a+");
//making a file ptr and opening a temp file with read and append mode
        ch=fgetc(fp1);
fprintf(fp,"Subject-Name=DLDA\t\tCOURSECODE=%s\tSubject-
Incharge=Apeksha\n",subjc1);
        printf("\nPress p for present and a for absent\n");
        while(ch!=EOF)
        {
                        if(ch==EOL)
                        do
                        {
                        p=getch();
                        }while(p!='p'&&p!='a');
                        if(p=='p')
                        {
                        while(ch!=EOF)
                        {
```

```
                    if(ch==EOL)
                    x++;
                    if(ch==EOL)
                    fprintf(fp,"\t\t%s",present);
                    if(ch==EOL)
                    printf(present);
                    break;
                }
                }
                else if(p=='a')
                {
                while(ch!=EOF)
                {
                    if(ch==EOL)
                    y++;
                    if(ch==EOL)
                    fprintf(fp,"\t\t%s",absent);
                    if(ch==EOL)
                    printf(absent);
                    break;
                }
                }
                fputc(ch, fp);

                    putchar(ch);
                ch = fgetc(fp1);
            }
printf("\nTotal number of student Present\t%d\ton %s at %s",x,date1,time1);
printf("\nTotal number of student Absent \t%d\ton %s at %s",y,date1,time1);
fprintf(fp,"\nTotal number of student Present\t%d\ton %s at %s",x,date1,time1);
fprintf(fp,"\nTotal number of student Absent \t%d\ton %s at %s",y,date1,time1);
            x=00,y=00;    //again reseting counter to 0
            fclose(fp);
            fclose(fp1);
```

```
labelitrl1:printf("\nPress y to commit or n to update the record\n");
                l1=getch();
                if(l1=='y')
                {
                fp=fopen(c1,"r");
                FILE *fp5;
                fp5=fopen(c11,"w");
                ch=fgetc(fp);
                while(ch!=EOF)
                {
                        fputc(ch,fp5);
                        ch=fgetc(fp);
                }
                fclose(fp);
                fclose(fp5);
                remove(c1);
                }
                else if(l1=='n')
                {
                        remove(c1);
                        goto labelitr;
                }
                else
                {
                        printf("\nPlease select only above mentioned option\n");
                                goto labelitrl1;
                }
                        rmdir(d1);
label4:printf("\nPress b to go back to main menu or press e to Logout\n");
                        ch4=getch();
                        if(ch4=='b')
                        {
                                goto label2;
                        }
```

```
            else if(ch4=='e')
            {
                goto x1;
            }
            else
            {
            printf("\nPlease select only above mentioned option\n");
                goto label4;
            }
        }
        else
        {
        printf("Invalid Subject Code , plz enter subject code in caps\n or it may be possible the subject is taken by other faculty\nTry Again");
            goto label81;
        }
}
        else if(ch2=='z')
        {
            do{
            DIR *d;
            struct dirent *dir;
            d=opendir("..//DSUMini/Faculty/Apeksha/");
            if(d)
            {
                printf("\nTotal records are in ur database\n");
                while((dir=readdir(d))!=NULL)
                {
                    printf("%s\n", dir->d_name);
                }
                closedir(d);

            }
```

```
        char filename[50];
        int f;
        char c1[]="..//DSUMini/Faculty/Apeksha/";
        printf("\nEnter the date and time in the following format
        DD-MM-YYYY-HH-AM/PM\nExample:  21-08-2017-
        08-AM To delete any specific records\n");
        scanf("%s",filename);
        strcat(c1,filename);
        char c2[]=".txt";
        strcat(c1,c2);
        strcat(filename,c2);
        FILE *fp2;
        fp2=fopen(c1,"r");
        if(fp2==NULL)
        {
        printf("\nNo such records found plz try again\npress b to
        go back or any key to contiue\n");
                ch6=getch();
                if(ch6=='b')
                {
                        goto label2;
                }
        }
fclose(fp2);

if(fp2!=NULL)
{
printf("Are You Sure You really want to delete %s \nRecord/File
From the database",filename);
        label5:printf("\nThe recovery isn't possible after deletion
        Press y to confirm or b to go back ");
        ch5=getch();
        if(ch5=='y')
        {
```

```
            f=remove(c1);
            if(f==0)
            {
            printf("\n%s Record/File deted successfully\npress b to go
            back or press any key to continue",filename);
                  ch8=getch();
                  if(ch8=='b')
                  {
                        goto label2;
                  }
            }
            else
            {
                  printf("Unable to delete file");
                  perror("Error");
            }
            }
else
{
            if(ch5=='b')
            {
                  goto label2;
            }
            else
            {
            printf("\nPlease select only above mentioned option\n");
                  goto label5;
            }
}
}
}while(fp2!=NULL);
}
else if(ch2=='x')
{
```

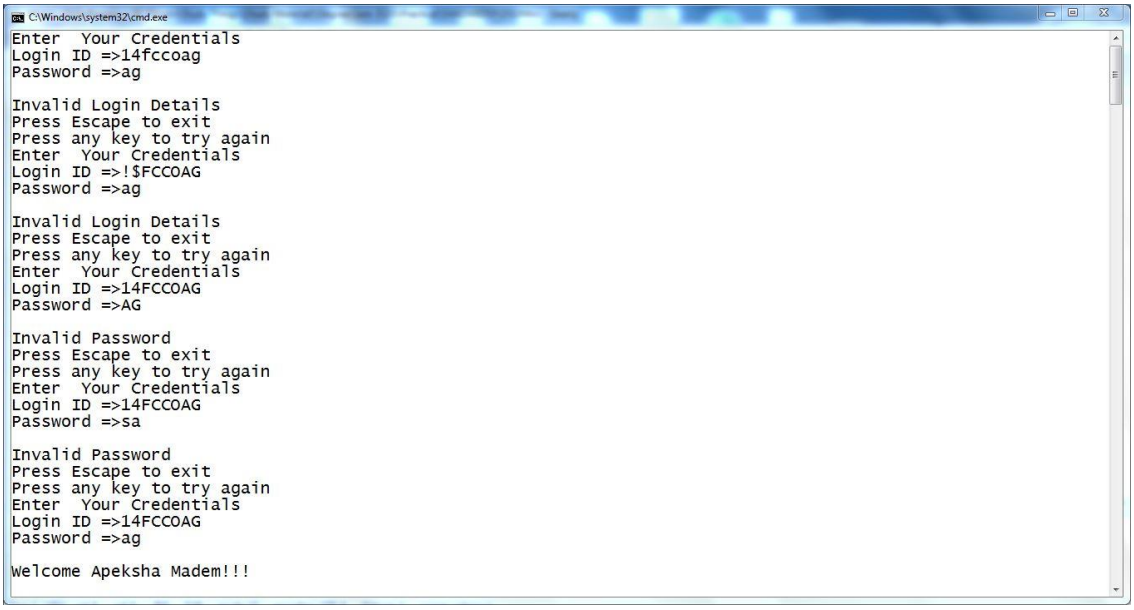
```
                goto x1;
            }
            else
            {
            printf("\nPlease select only the above option mentioned\n");
                goto label1;
            }
        }
        else
        {
    printf("\nInvalid Password\nPress Escape to exit\nPress any key to try again\n");
            chx=getch();
            if(chx==esc)
            {
                return 0;
            }
            else
            {
                goto x1;
            }
        }
    }
}
```

# Chapter 4

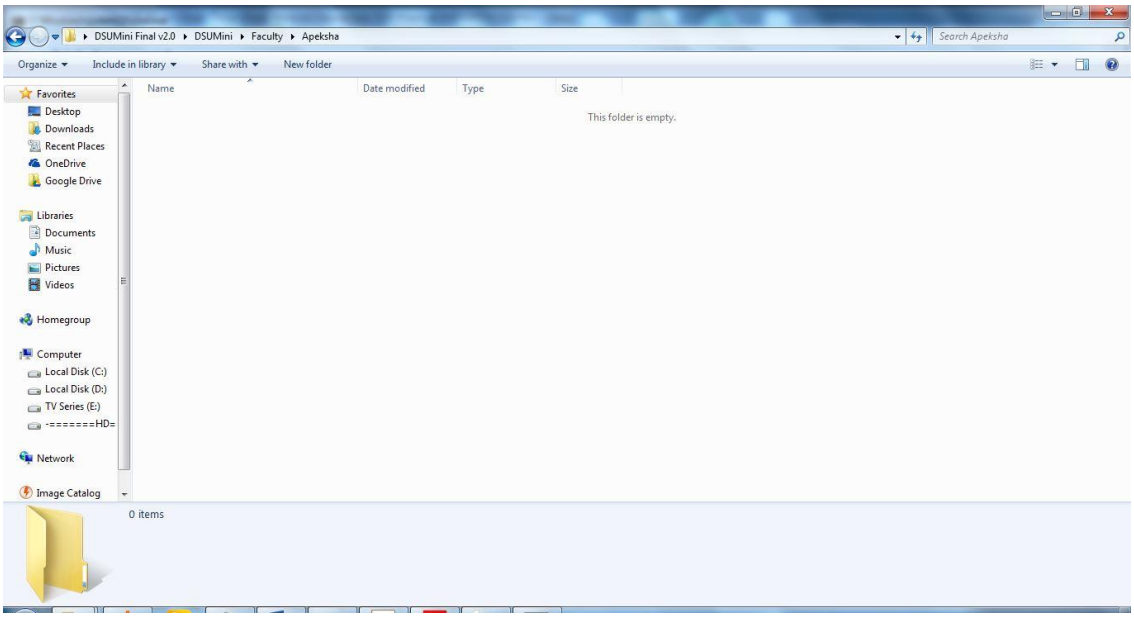
## RESULTS

# 4.1 Screenshots

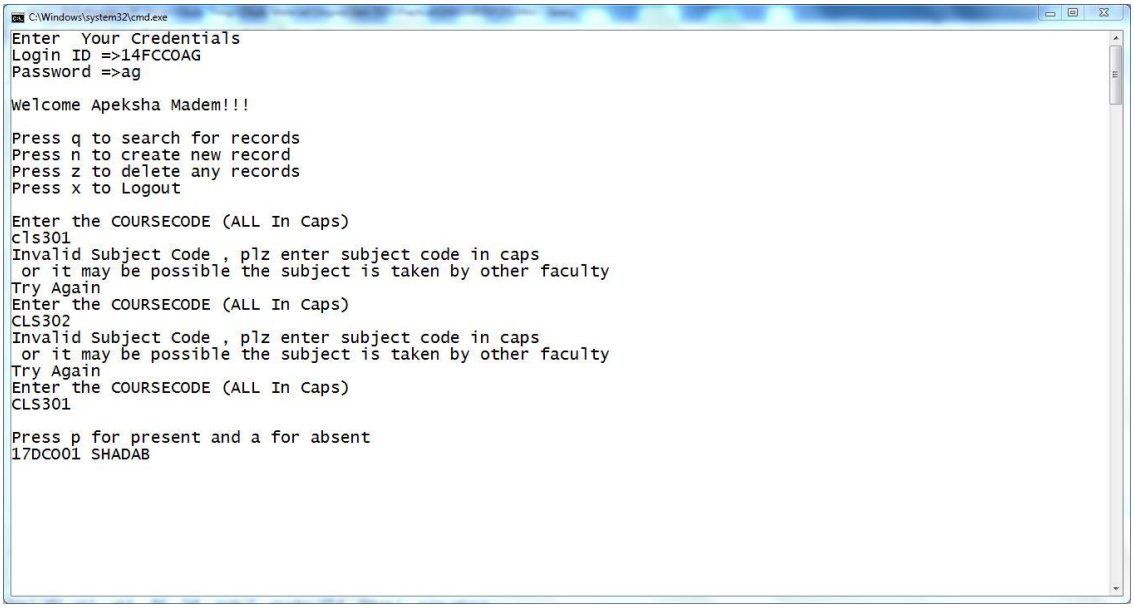
## User Entering invalid details and valid details



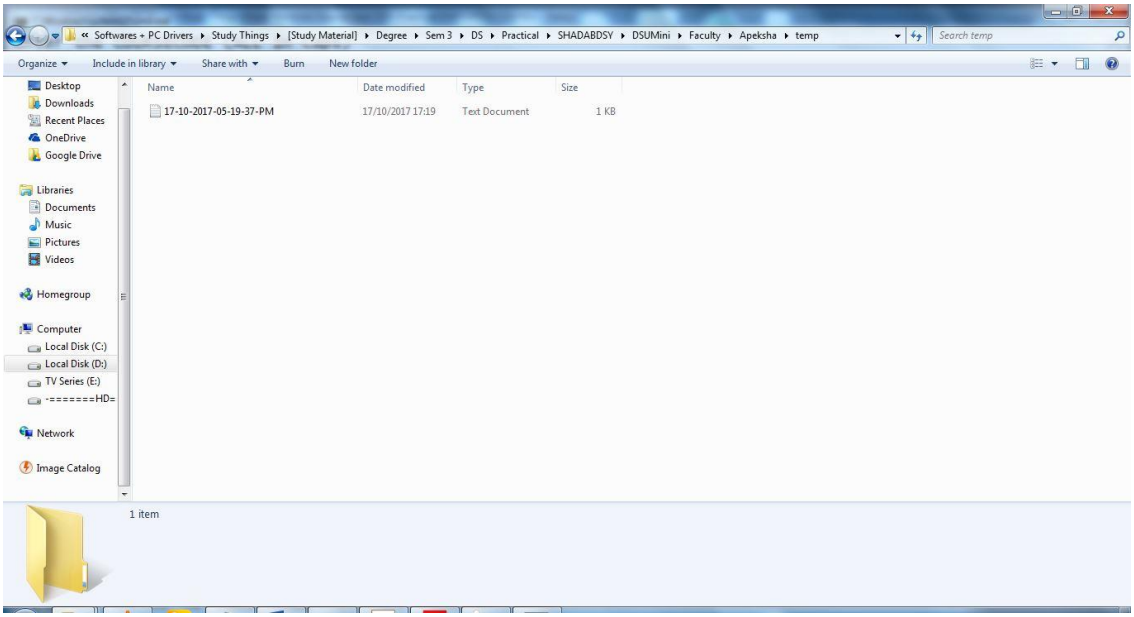

## Accessing the database (No files currently)



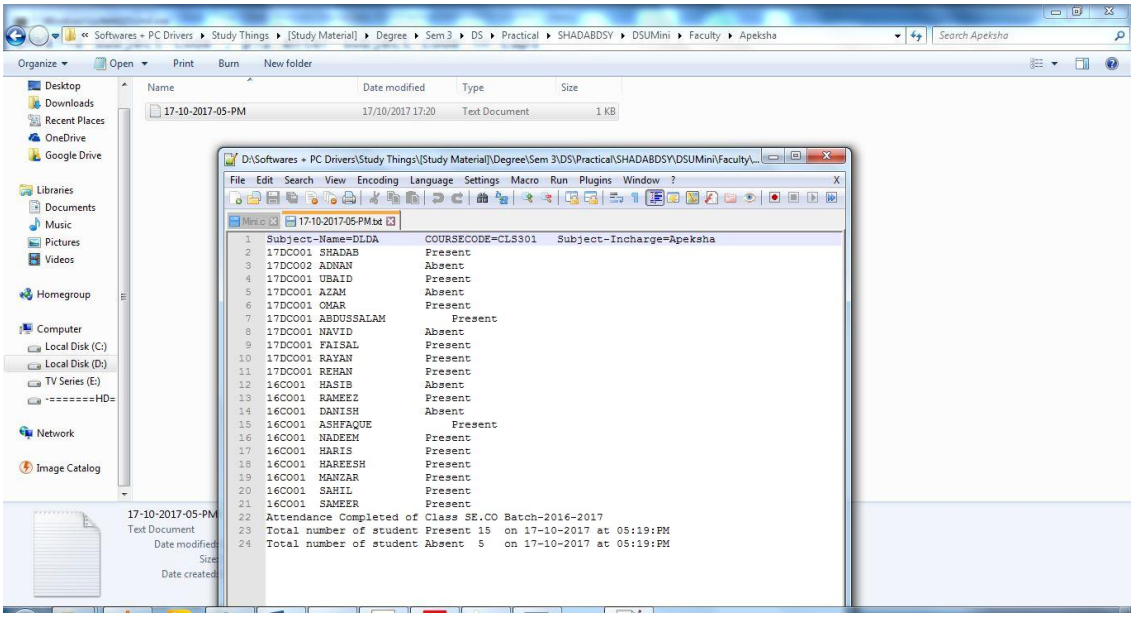
## Creating a new record (pressing n and entering invalid course code and valid course code)



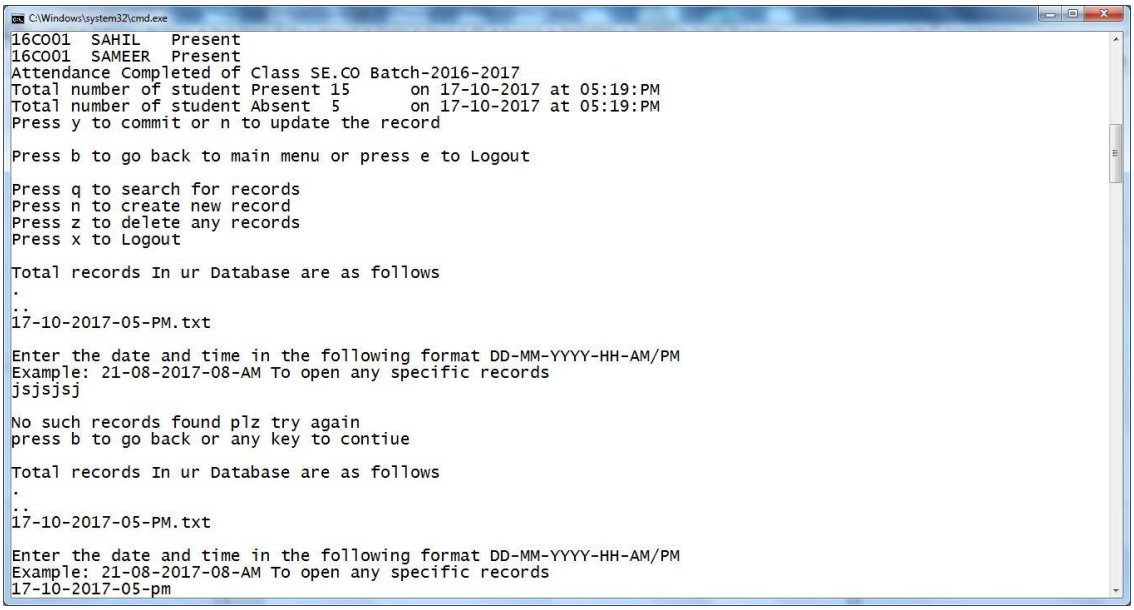

## Files being created in temp folder



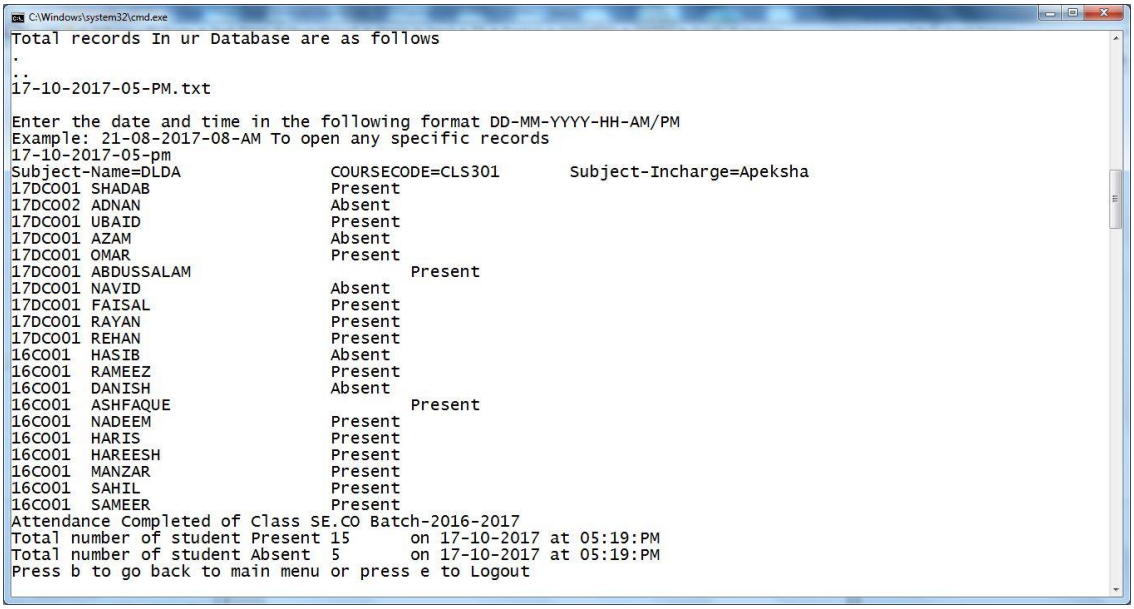
## Files being saved permanently folder in permanent database



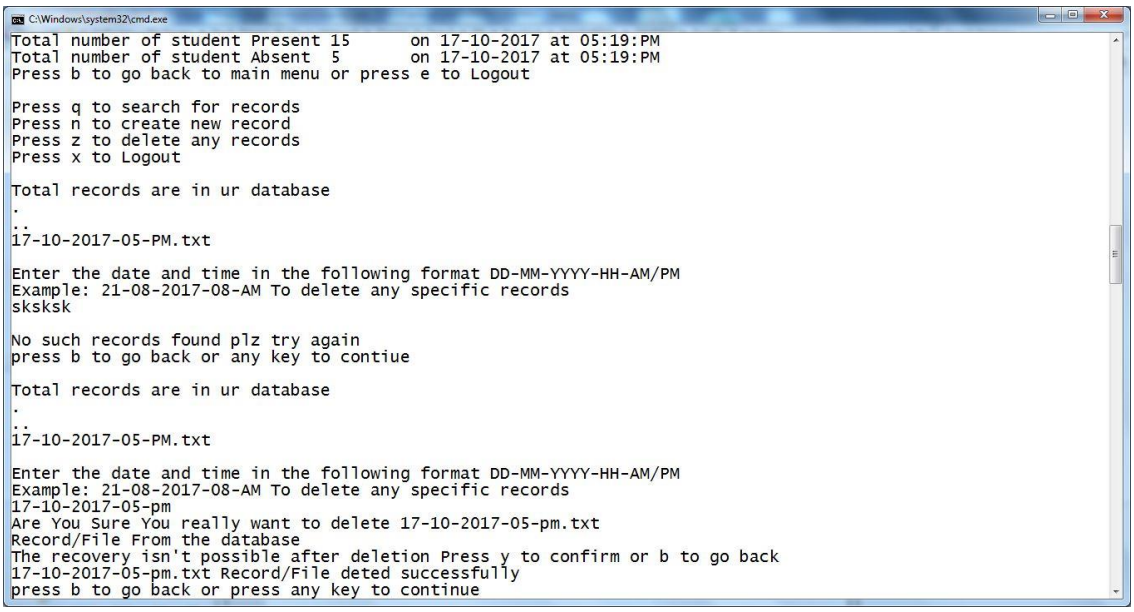

## Going back to main menu, reading records, user able to see files in their database (entering invalid file name and then valid name)



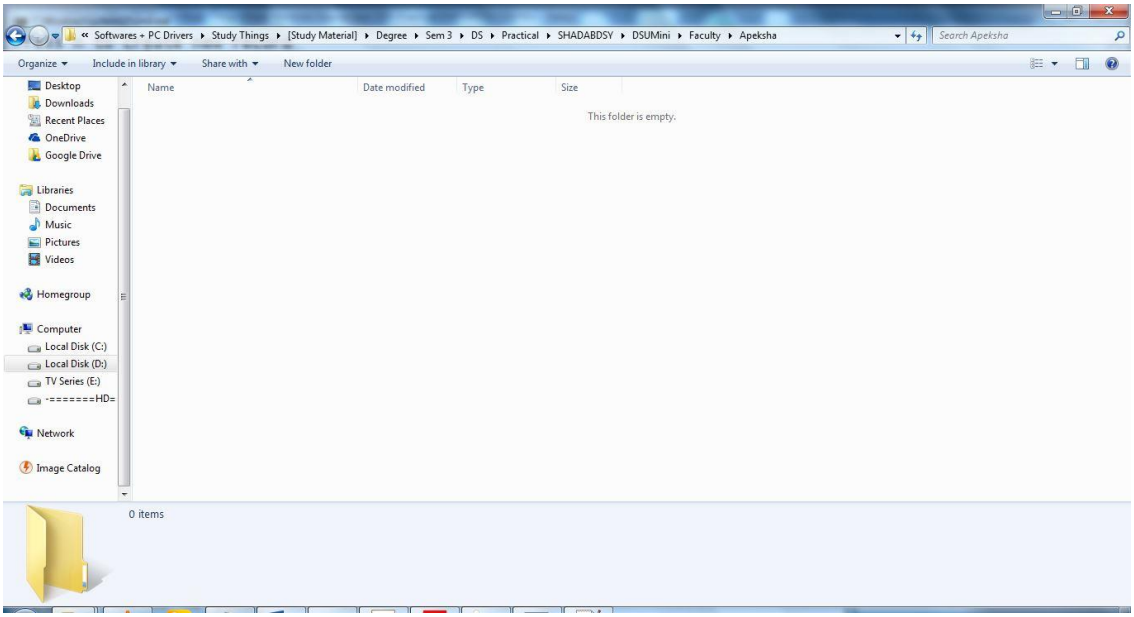
## Reading full records



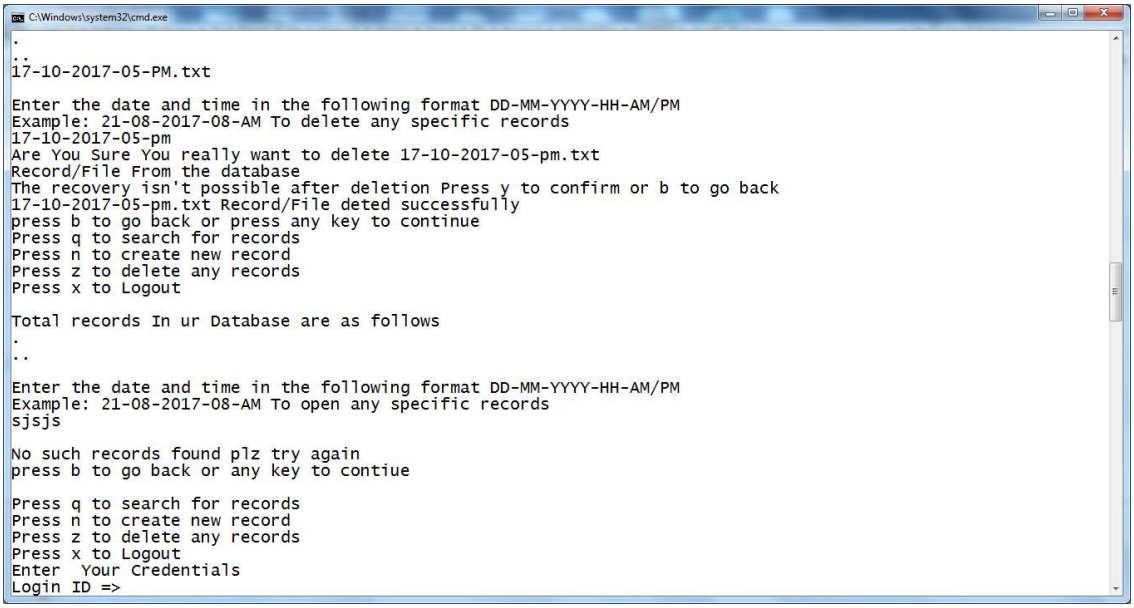

## Going back to main menu, deleting record, (entering invalid name and then valid name, asking if sure to delete)



# File deleted from database




# Logging Out.



# Chapter 5

## Conclusion

## 5.1 Conclusion

By implementing this project, we have successfully overcome some of the drawbacks of the traditional mechanism of taking attendance.

We have also ensured that a particular file safe secured and reliable for different faculties. Making the file permanent electronically saving in a txt file.

With this project we have got a basic idea of how FILE structure works and array of structure, and how to deal with file in C such as how to open a file in various modes (read, write, append, append+ - for reading and appending).

We have also used various libraries and learned how to implement those such as time.h and dirent.h for creation of time and directory and also used strftime method available under time.h library for formatting of time.

This project can be enhanced by implementing some more modules like making a graph of attendance of student, single student details fetching, making a defaulter list.

# References

[1] StackOverflow.com, 'getting-date-and-time-in-c', 2012. [Online]. Available: https://stackoverflow.com/questions/9233123/getting-date-and-time-in-c.html [Accessed: 19- Oct- 2017].

[2] sanfoundry.com, 'c-program-illustrate-reading-datafile', 2012. [Online]. Available: http://www.sanfoundry.com/c-program-illustrate-reading-datafile/ [Accessed: 19- Oct- 2017].

[3] tutorialspoint.com, 'c_function_strftime', 2012. [Online]. Available: https://www.tutorialspoint.com/c_standard_library/c_function_strftime.htm [Accessed: 19- Oct- 2017].

[4] c4learn.com, 'c-file-open-modes', 2013. [Online]. Available: http://www.c4learn.com/c-programming/c-file-open-modes/ [Accessed: 19- Oct- 2017].