
/*****

Application: Record Entry System
Compiled on: Borland Turbo C++ 3.0
Programmer: Geetika Mukhi

*****/

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
#include <dos.h>
#include <ctype.h>
```

```
void dataentry(void);
void selectAdminOption(void);
void getData(int option);
int showAdminMenu;
```

```
void main()
```

```
{
    int cancelOption,timeOption,entryOption,exitOption;
    char choice[1];
    char selectOption[1];

    textcolor(YELLOW);
    cancelOption=0;

    /* Shows the main menu for the application*/
    while (cancelOption==0)
    {
        clrscr();
        gotoxy(30,7);
        printf("Please Select an Action-->");
        gotoxy(30,10);
        printf("Daily Time Record [1] ");
        gotoxy(30,11);
        printf("Data Entry          [2] ");
        gotoxy(30,12);
        printf("Close                [3] ");
        gotoxy(30,15);
        printf("Please Enter Your Choice (1/2/3): ");
        scanf("%s",&choice);
        timeOption=strcmp(choice,"1");
        entryOption=strcmp(choice,"2");
        exitOption=strcmp(choice,"3");

        if (timeOption==0)
        {
            clrscr();
            gotoxy(23,6);
            printf("DAILY EMPLOYEE TIME RECORDING SYSTEM");
            gotoxy(16,24);
            printf("Input Any Other key to Return to Previous Screen.");
            gotoxy(31,9);
            printf("[1] Employee Log In ");
            gotoxy(31,10);
            printf("[2] Employee Log Out");
            gotoxy(28,12);
            printf("Please Enter Your Option: ");
            scanf("%s",&selectOption);
            if (strcmp(selectOption,"1")==0)
            {
                getData(5);
            }
            if (strcmp(selectOption,"2")==0)
            {

```

```

        getData(6);
    }
    cancelOption=0;
}
if (entryOption==0)
{
    dataentry();
    cancelOption=0;
}
if (exitOption==0)
{
    cancelOption=1;
}

if (!(timeOption==0 || entryOption==0 || exitOption==0))
{
    gotoxy(10,17);
    printf("You Have Entered an Invalid Option. Please Choose Either 1, 2 or 3. ");
};
    getch();
    cancelOption=0;
}

}
clrscr();
gotoxy(23,13);
printf("The Application will Close Now. Thanks!");
getch();
}

```

/* This function provides logic for data entry to be done for the system.
Access to Data Entry screens will be only allowed to administrator user.*/

```

void dataentry(void)
{
    char adminName[10], passwd[5],buffer[1];
    char tempo[6],sel[1];
    int validUserNameOption,validUserPwdOption,returnOption,UserName,inc,tmp;
    char plus;
    clrscr();
    validUserNameOption=0;
    validUserPwdOption=0;
    while (validUserPwdOption==0)
    {
        clrscr();
        while (validUserNameOption==0)
        {
            clrscr();
            gotoxy(20,5);
            printf("IT SOFTWARE DATA ENTRY SYSTEM-ADMIN INTERFACE");
            gotoxy(20,24);
            printf("Info: Type return to go back to the main screen.");
            gotoxy(28,10);
            printf("Enter Administrator Name: ");
            scanf("%s",&adminName);
            returnOption=strcmp(adminName,"return");
            UserName=strcmp(adminName,"admin");

            if (returnOption==0)
            {
                goto stream;
            }
            if (!(UserName==0 || returnOption==0))
            {

                gotoxy(32,11);
                printf("Administrator Name is Invalid.");
                getch();
            }
        }
    }
}

```

```

        validUserNameOption=0;
    }
    else
        validUserNameOption=1;
}

gotoxy(30,11);
printf("Enter Password: ");
inc=0;
while (inc<5)
{
    passwd[inc]=getch();
    inc=inc+1;
    printf("* ");
}
inc=0;
while (inc<5)
{
    tempo[inc]=passwd[inc];
    inc=inc+1;
}
while(getch()!=13);
if (!strcmp(tempo, "admin12"))
{
    gotoxy(28,13);
    printf("You have Entered a Wrong Password. Please Try Again. ");
    getch();
    validUserPwdOption=0;
    validUserNameOption=0;
}
else
{
    clrscr();
    gotoxy(24,11);
    textcolor(YELLOW+BLINK);
    cprintf("You Have Successfully Logged In.");
    gotoxy(24,17);
    textcolor(YELLOW);
    printf("Press Any Key to Continue.");
    validUserPwdOption=1;
    validUserNameOption=1;
    getch();
    showAdminMenu=0;

    while (showAdminMenu==0)
    {
        clrscr();
        gotoxy(24,4);
        printf("ADMIN OPTIONS");
        gotoxy(26,9);
        printf("Add New Employee      [1]");
        gotoxy(26,11);
        printf("Show Daily Entries      [2]");
        gotoxy(26,13);
        printf("Search Employee Record [3]");
        gotoxy(26,15);
        printf("Remove Employee         [4]");
        gotoxy(26,17);
        printf("Close                    [5]");
        gotoxy(24,21);
        printf("Please enter your choice: ");
        selectAdminOption();
    }
}
}
stream: {}

```

```

}

/* This function provides the administrator level functionalities, such as Adding or
deleting an employee.*/
void selectAdminOption(void)
{
    char chc[1];
    int chooseNew,chooseShow,chooseSearch,chooseRemove,chooseClose;

    gets(chc);

    chooseNew=strcmp(chc,"1");
    chooseShow=strcmp(chc,"2");
    chooseSearch=strcmp(chc,"3");
    chooseRemove=strcmp(chc,"4");
    chooseClose=strcmp(chc,"5");

    if (!(chooseNew==0 || chooseShow==0 || chooseSearch==0 || chooseRemove==0 ||
chooseClose==0))
    {
        gotoxy(19,21);
        textcolor(RED+BLINK);
        cprintf("Invalid Input!");
        gotoxy(34,21);
        textcolor(YELLOW);
        cprintf("Press any key to continue.");
    }

    if (chooseNew==0)
    {
        clrscr();
        gotoxy(25,5);
        getData(1);
    }
    else if(chooseShow==0)
    {
        getData(2);
    }
    else if(chooseSearch==0)
    {
        clrscr();
        getData(3);
    }
    else if(chooseRemove==0)
    {
        getData(4);
    }
    else if (chooseClose==0)
    {
        showAdminMenu=1;
    }
}

/* This function retrieves data from the database as well as do data processing
according to user requests.
The function provides functionality for menu options provided to both employee as
well as administrator user*/
void getData(int option)
{
    FILE *db,*tempdb;
    char anotherEmp;
    int choice;
    int showMenu,posx,posy;
    char checkSave,checkAddNew;
    int i;

    struct employee
    {

```

```

    char firstname[30];
    char lastname[30];
    char password[30];
    int empid;
    char loginhour;
    char loginmin;
    char loginsec;
    char logouthour;
    char logoutmin;
    char logoutsec;
    int yr;
    char mon;
    char day;
};

struct employee empData;

char confirmPassword[30];
long int size;
char lastNameTemp[30], firstNameTemp[30], password[30];
int searchId;
char pass[30];
char findEmployee;
char confirmDelete;

struct date today;
struct time now;

clrscr();

/* Opens the Employee Database*/
db=fopen("d:/empbase.dat", "rb+");
if(db==NULL)
{
    db=fopen("d:/empbase.DAT", "wb+");
    if(db==NULL)
    {
        printf("The File could not be opened.\n");
        exit();
    }
}
printf("Application Database \n");
size=sizeof(empData);
showMenu=0;
while(showMenu==0)
{
    fflush(stdin);
    choice=option;

    /* Based on the choice selected by admin/employee, this switch statement processes
the request*/
    switch(choice)
    {

        /* To add a new employee to the database*/
        case 1:
            fseek(db, 0, SEEK_END);
            anotherEmp='y';

            while(anotherEmp=='y')
            {
                checkAddNew=0;
                while(checkAddNew==0)
                {
                    clrscr();
                    gotoxy(25, 3);
                    printf("ADD A NEW EMPLOYEE");
                    gotoxy(13, 22);

```

```

printf("Warning: Password Must Contain Six(6) AlphaNumeric Digits.");
gotoxy(5,8);
printf("Enter First Name: ");
scanf("%s",&firstNameTemp);
gotoxy(5,10);
printf("Enter Last Name: ");
scanf("%s",&lastNameTemp);
gotoxy(43,8);
printf("Enter Password: ");
for (i=0;i<6;i++)
{
    password[i]=getch();
    printf("* ");
}
password[6]='\0';

while(getch()!=13);

gotoxy(43,10);
printf("Confirm Password: ");
for (i=0;i<6;i++)
{
    confirmPassword[i]=getch();
    printf("* ");
}
confirmPassword[6]='\0';

while(getch()!=13);
if (strcmp(password,confirmPassword))
{
    gotoxy(24,12);
    printf("Passwords do not match.");
    gotoxy(23,13);
    printf("Press any key to continue.");
    getch();
}
else
{
    checkAddNew=1;
    rewind(db);
    empData.empid=0;
    while(fread(&empData,size,1,db)==1);
    if (empData.empid<2000)
        empData.empid=20400;

    empData.empid=empData.empid+1;
    gotoxy(29,16);
    printf("Save Employee Information? (y/n): ");
    checkSave=getche();
    if (checkSave=='y')
    {
        strcpy(empData.firstname,firstNameTemp);
        strcpy(empData.lastname,lastNameTemp);
        strcpy(empData.password,password);
        empData.loginhour='t';
        empData.logouthour='t';
        empData.day='j';
        fwrite(&empData,size,1,db);
    }
    gotoxy(28,16);
    printf(" ");
    gotoxy(28,16);
    printf("would like to add another employee? (y/n):");
    fflush(stdin);
    anotherEmp=getche();
    printf("\n");
}
}
}

```

```

    }
    break;

/* To view time records for all employees*/
case 2:

    clrscr();
    gotoxy(21,2);
    printf("VIEW EMPLOYEE INFORMATION");
    gotoxy(1,5);
    printf("Employee ID    Employee Name        Time Logged In        Time Logged Out\n");
    rewind(db);
    posx=3;
    posy=7;
    while(fread(&empData,size,1,db)==1)
    {
        empData.firstname[0]=toupper(empData.firstname[0]);
        empData.lastname[0]=toupper(empData.lastname[0]);
        gotoxy(posx,posy);
        printf("%d",empData.empid);
        gotoxy(posx+10,posy);
        printf("| %s, %s",empData.lastname,empData.firstname);
        gotoxy(posx+30,posy);
        if (empData.loginhour=='t')
        {
            printf("| Not Logged In");
        }
        else
            printf("| %d:%d:%d",empData.loginhour,empData.loginmin,empData.loginsec);

        gotoxy(posx+49,posy);
        if (empData.logouthour=='t')
        {
            printf("| Not Logged Out");
        }
        else
            printf("| %d:%d:%d",empData.logouthour,empData.logoutmin,empData.logoutsec);
        if (empData.day=='j')
        {
            gotoxy(posx+69,posy);
            printf("| No Date");
        }
        else
        {
            gotoxy(posx+73,posy);
            printf("| %d/%d/%d",empData.mon,empData.day,empData.yr);
        }

        posy=posy+1;
    }
    getch();

    printf("\n");
    break;

/* To search a particular employee and view their time records*/
case 3:

    clrscr();
    gotoxy(27,5);
    printf("SEARCH EMPLOYEE INFORMATION");
    gotoxy(25,9);
    printf("Enter Employee Id to Search: ");
    scanf("%d", &searchId);
    findEmployee='f';
    rewind(db);
    while(fread(&empData,size,1,db)==1)

```

```

{
    if (empData.empid==searchId)
    {
        gotoxy(33,11);
        textcolor(YELLOW+BLINK);
        cprintf("Employee Information is Available.");
        textcolor(YELLOW);
        gotoxy(25,13);
        printf("Employee name is: %s %s",empData.lastname,empData.firstname);
        if(empData.loginhour=='t')
        {
            gotoxy(25,14);
            printf("Log In Time: Not Logged In");
        }
        else
        {
            gotoxy(25,14);
            printf("Log In Time is: %d:%d:%d",empData.loginhour,empData.loginmin,empData.
loginsec);
        }
        if(empData.logouthour=='t')
        {
            gotoxy(25,15);
            printf("Log Out Time: Not Logged Out");
        }
        else
        {
            gotoxy(25,15);
            printf("Log Out Time is: %d:%d:%d",empData.logouthour,empData.logoutmin,empData.
logoutsec);
        }
        findEmployee='t';
        getch();
    }
}
if (findEmployee!='t')
{
    gotoxy(30,11);
    textcolor(YELLOW+BLINK);
    cprintf("Employee Information not available. Please modify the search.");
    textcolor(YELLOW);
    getch();
}
break;

/* To remove entry of an employee from the database*/
case 4:
    clrscr();
    gotoxy(25,5);
    printf("REMOVE AN EMPLOYEE");
    gotoxy(25,9);
    printf("Enter Employee Id to Delete: ");
    scanf("%d", &searchId);
    findEmployee='f';
    rewind(db);
    while(fread(&empData,size,1,db)==1)
    {
        if (empData.empid==searchId)
        {
            gotoxy(33,11);
            textcolor(YELLOW+BLINK);
            cprintf("Employee Information is Available.");
            textcolor(YELLOW);
            gotoxy(25,13);
            printf("Employee name is: %s %s",empData.lastname,empData.firstname);
            findEmployee='t';

```



```

    }
}
if (findEmployee!='t')
{
gotoxy(30,11);
textcolor(YELLOW+BLINK);
cprintf("Employee Information not available. Please modify the search.");
textcolor(YELLOW);
getch();
}
if (findEmployee=='t')
{
gotoxy(29,15);
printf("Do you want to Delete the Employee? (y/n)");
confirmDelete=getche();
if (confirmDelete=='y' || confirmDelete=='Y')
{
tempdb=fopen("d:/tempo.dat","wb+");
rewind(db);
while(fread(&empData,size,1,db)==1)
{
if (empData.empid!=searchId)
{
fseek(tempdb,0,SEEK_END);
fwrite(&empData,size,1,tempdb);
}
}
fclose(tempdb);
fclose(db);
remove("d:/empbase.dat");
rename("d:/tempo.dat","d:/empbase.dat");
db=fopen("d:/empbase.dat","rb+");
}
}
break;

/* To login an employee into the system and record the login date and time*/
case 5:
clrscr();
gotoxy(20,4);
printf("DAILY EMPLOYEE TIME RECORDING SYSTEM");
gotoxy(20,23);
printf("Warning: Please Enter Numeric Values Only.");
gotoxy(23,7);
printf("Enter Your Id to Login: ");
scanf("%d", &searchId);
gotoxy(20,23);
printf("                                ");
findEmployee='f';
rewind(db);
while(fread(&empData,size,1,db)==1)
{
if (empData.empid==searchId)
{
gotoxy(23,8);
printf("Enter Your Password: ");

for (i=0;i<6;i++)
{
pass[i]=getch();
printf("* ");
}
pass[6]='\0';
while(getch()!=13);

if (strcmp(empData.password,pass))
{
gotoxy(23,11);

```

```

        textcolor(YELLOW+BLINK);
        cprintf("You Have Supplied a Wrong Password.");
        textcolor(YELLOW);
        findEmployee='t';
        getch();
        break;
    }
    gotoxy(23,11);
    textcolor(YELLOW+BLINK);
    cprintf("You have successfully Logged In the System.");
    textcolor(YELLOW);
    gotoxy(23,13);
    printf("Employee name: %s %s",empData.lastname,empData.firstname);
    gettime(&now);
    getdate(&today);
    gotoxy(23,14);
    printf("Your LogIn Time: %2d:%2d:%2d",now.ti_min,now.ti_hour,now.ti_sec);
    gotoxy(23,15);
    printf("Your Log In Date: %d/%d/%d",today.da_mon,today.da_day,today.da_year);
    empData.day=today.da_day;
    empData.mon=today.da_mon;
    empData.yr=today.da_year;
    fseek(db,-size,SEEK_CUR);
    empData.loginhour=now.ti_min;
    empData.loginmin=now.ti_hour;
    empData.loginsec=now.ti_sec;
    fwrite(&empData,size,1,db);
    findEmployee='t';
    getch();

    }
}
if (findEmployee!='t')
{
    gotoxy(30,11);
    textcolor(YELLOW+BLINK);
    cprintf("Employee Information is not available.");
    textcolor(YELLOW);
    getch();
}

break;

```

/* To logout an employee and record the logout date and time*/
case 6:

```

    clrscr();
    gotoxy(20,4);
    printf("DAILY EMPLOYEE TIME RECORDING SYSTEM");
    gotoxy(20,23);
    printf("Warning: Please Enter Numeric Values Only.");
    gotoxy(23,7);
    printf("Enter Your Id to Logout: ");
    scanf("%d", &searchId);
    gotoxy(20,23);
    printf(" ");
    findEmployee='f';
    rewind(db);
    while(fread(&empData,size,1,db)==1)
    {
        if (empData.empid==searchId)
        {
            gotoxy(23,8);
            printf("Enter Password: ");

            for (i=0;i<6;i++)
            {
                pass[i]=getch();
            }
        }
    }
}

```

```
        printf("* ");
    }
    pass[6]='\0';
    while(getch() != 13);

    if (strcmp(empData.password, pass))
    {
        gotoxy(30, 11);
        textcolor(YELLOW+BLINK);
        cprintf("You Have Supplied a wrong Password.");
        textcolor(YELLOW);
        findEmployee='t';
        getch();
        break;
    }
    gotoxy(23, 11);
    textcolor(YELLOW+BLINK);
    cprintf("You have successfully Logged Out of the System.");
    textcolor(YELLOW);
    gotoxy(23, 13);
    printf("Employee name is: %s %s", empData.lastname, empData.firstname);
    gettimeofday(&now);
    gettimeofday(&today);
    gotoxy(23, 14);
    printf("Your Log Out Time: %d:%d:%d", now.ti_min, now.ti_hour, now.ti_sec);
    gotoxy(23, 15);
    printf("Your Log Out Date: %d/%d/%d", today.da_mon, today.da_day, today.da_year);
    fseek(db, -size, SEEK_CUR);
    empData.logouthour=now.ti_min;
    empData.logoutmin=now.ti_hour;
    empData.logoutsec=now.ti_sec;
    fwrite(&empData, size, 1, db);
    findEmployee='t';
    getch();

    }
}
if (findEmployee != 't')
{
    gotoxy(23, 11);
    textcolor(YELLOW+BLINK);
    cprintf("Employee Information is not available.");
    textcolor(YELLOW);
    getch();
}

break;

/* show previous menu*/
case 9:

    printf("\n");
    exit();
}
fclose(db);
showMenu=1;
}
}
```

Please Select an Action-->

Daily Time Record	[1]
Data Entry	[2]
Close	[3]

Please Enter Your Choice (1/2/3):

IT SOFTWARE DATA ENTRY SYSTEM-ADMIN INTERFACE

Enter Administrator Name: _

Info: Type return to go back to the main screen.

You Have Successfully Logged In.

Press Any Key to Continue._

ADMIN OPTIONS

- Add New Employee [1]
- Show Daily Entries [2]
- Search Employee Record [3]
- Remove Employee [4]
- Close [5]

Please enter your choice: _

ADD A NEW EMPLOYEE

Enter First Name: Peter

Enter Password: * * * * *

Enter Last Name: Jones

Confirm Password: * * * * *

Save Employee Information? (y/n): _

Warning: Password Must Contain Six(6) AlphaNumeric Digits.

ADD A NEW EMPLOYEE

Enter First Name: Peter

Enter Password: * * * * *

Enter Last Name: Jones

Confirm Password: * * * * *

Would like to add another employee? (y/n):

Warning: Password Must Contain Six(6) AlphaNumeric Digits.

VIEW EMPLOYEE INFORMATION

Employee ID	Employee Name	Time Logged In	Time Logged Out	Date
20401	Taylor, Frank	10:50:97	10:22:5	6/7
/220402	Smith, Annie	Not Logged In	Not Logged Out	No Date
20403	Jones, Peter	Not Logged In	Not Logged Out	No Date

SEARCH EMPLOYEE INFORMATION

Enter Employee Id to Search: _

SEARCH EMPLOYEE INFORMATION

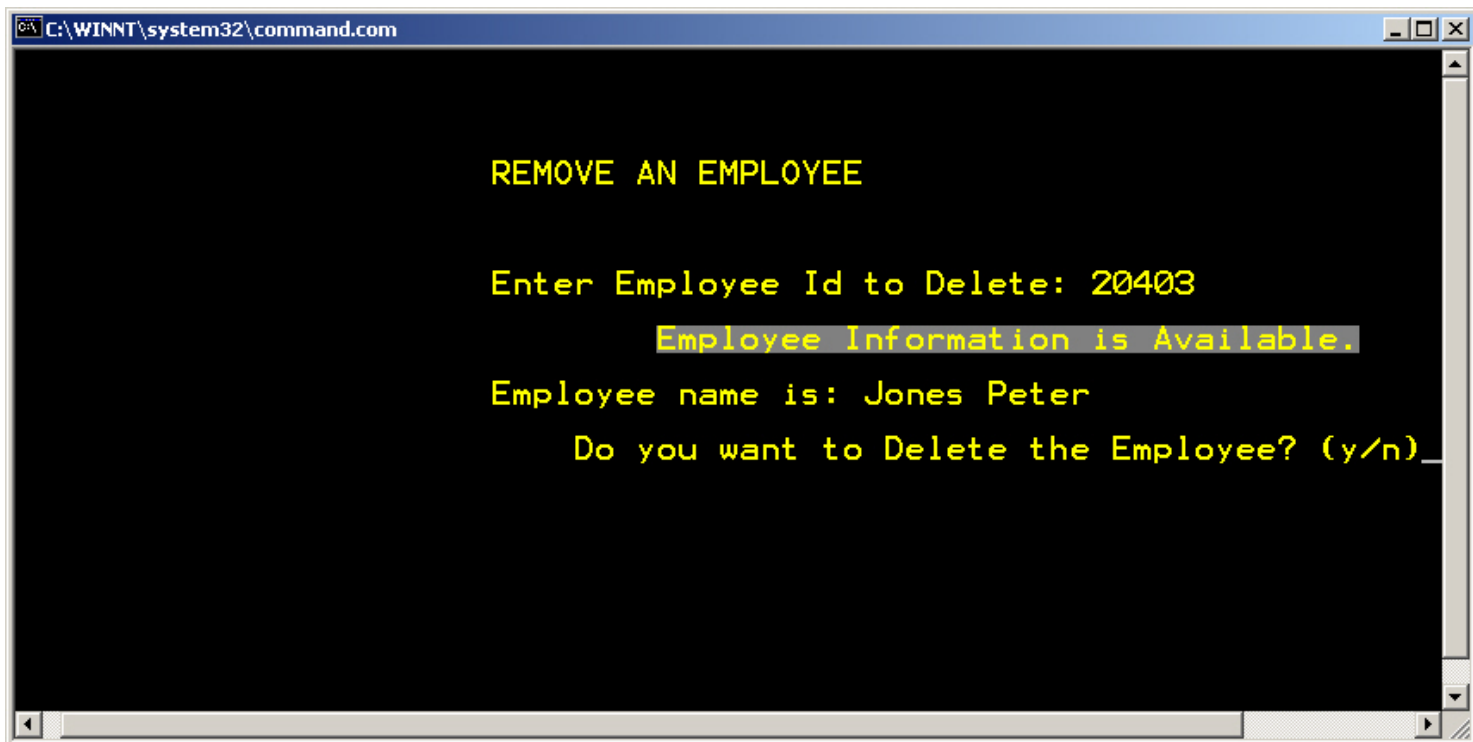
Enter Employee Id to Search: 20403

Employee Information is Available.

Employee name is: Jones Peter

Log In Time: Not Logged In

Log Out Time: Not Logged Out



```
C:\WINNT\system32\command.com

REMOVE AN EMPLOYEE

Enter Employee Id to Delete: 20403
    Employee Information is Available.
Employee name is: Jones Peter
    Do you want to Delete the Employee? (y/n)
```

DAILY EMPLOYEE TIME RECORDING SYSTEM

- [1] Employee Log In
- [2] Employee Log Out

Please Enter Your Option:

Input Any Other key to Return to Previous Screen.

DAILY EMPLOYEE TIME RECORDING SYSTEM

Enter Your Id to Login:

Warning: Please Enter Numeric Values Only.

DAILY EMPLOYEE TIME RECORDING SYSTEM

Enter Your Id to Login: 20403
Enter Your Password: * * * * *

DAILY EMPLOYEE TIME RECORDING SYSTEM

Enter Your Id to Login: 20403
Enter Your Password: * * * * *

You have successfully Logged In the System.

Employee name: Jones Peter
Your LogIn Time: 3:12:39
Your Log In Date: 6/8/2007_

DAILY EMPLOYEE TIME RECORDING SYSTEM

Enter Your Id to Logout: _

Warning: Please Enter Numeric Values Only.

DAILY EMPLOYEE TIME RECORDING SYSTEM

Enter Your Id to Logout: 20403
Enter Password: * * * * *

You have successfully Logged Out of the System.

Employee name is: Jones Peter
Your Log Out Time: 4:12:39
Your Log Out Date: 6/8/2007_

