

Sandip Institute of Technology And Research Centre, Nashik  
Department of Computer Engineering (2020-21)

**SAVITRIBAI PHULEPUNE UNIVERSITY**

**A MINI PROJECT REPORT ON**

**“Password Protected Personal Dairy”**

Submitted by Group Id BE2020A11:

1. Sharvari Birute.
2. Chetana Mahajan.
3. Hemal Sawdekar

Under The Guidance of

Prof. Vivek Waghmare



DEPARTMENT OF COMPUTER ENGINEERING

**Sandip Institute of Technology And Research Centre, Nashik  
Mahiravani, Trimbak Road, Tal Dist. Nashik – 422 213,  
Maharashtra, India.**

## Acknowledgement

It is my immense pleasure to work on this preliminary project report on “ **Password Protected Personal Dairy**”.

I would like to take this opportunity to thank my internal guide **Prof.Vivek Waghmare sir** for giving me all help and guidance .I needed . I am really grateful to them for their kind of support .Their valuable suggestion were very helpful.

I am also grateful to **Prof.Amol Potgantwar sir** .Head of computer Engineering Department,Sandip Institute Of Technology And research center for his indispensable support and suggestion.

Sharvari Birute.  
Chetana Mahajan.  
Hemal Sawdekar  
(B.E Computer Engg.)

## Abstract

Password protected personal dairy is console application without graphics .In this project ,use Can keep their Personal record like they do in a dairy .you can keep record of important things you do in your daily life , like mettings and various other tasks .you can organize your schedule as well as view some particular entry of a particular date or time .In this console application you can add view , edit view and delete records . Records can be added with information such as duration of task , name address, time and date. Storing capacity is unlimited in records(Depends on memory).Its oops concept used on this password protected personal dairy.

<b>Sr.No</b>	<b>Name of topic</b>	<b>Page No.</b>
1	Introduction	5
2.	Introduction	5
3.	Overview	6
4.	Methodology	7
5.	Scope of Project	7
6.	Proposed system	8
7.	Advantages over Traditional System	8
8.	Specific Requirement	9
9.	Block Diagram	9
10.	Hardware Requirement	10
11.	Software Requirement	10
12.	Programming coding	11-19
13.	Output Screen	20
14.	Conclusion	21

## INTRODUCTION

Password Protected Personal Dairy project in C++.It is a console application without graphics .It based on concept to generate personal Daily Records and add records and to add record & update it. Here user can add their daily details safely and it's not time consuming. Each section contain high security so there is no chance of data loss .There maximum privacy to the record files. This System make easy to store daily records.The whole project is designed in 'C' language and different variables and strings have been used have been used for the development of this project. This mini project is easy to operator and understand by the user .This is simple mini project in C,built as a console application And features .It's easy to operate and understands by users .There is no any error and warning contents in this project .The design is so simple that user won't find it difficult to use and navigate. A personal dairy may include a person's experiences , which hiding from others and thoughts or feelings, including comment on current events outside the writer's direct experience. Dairies undertaken for institutional purposes plays role in many aspects of human civilization,including government records. The term dairy is some time called personal dairies ,you can add , view, edit, and delete records. Records can be added with information such as durations of task , name, address, time and date.

## Overview

Personal Diary Management System is based on concept to generate Personal Daily Records and to add records & update it.

The basic user-defined functions used in this project are listed below:

1. int password() –  
contains/manages/handles password protection
2. void add\_record() –  
to add a new diary record.
3. void view\_record() –  
to view added record in list.
4. void modify\_record() –  
to modify and update an added record.
5. void edit\_password() –  
to modify/change a password.
6. void delete\_record() –  
to delete or remove a record permanently from system file.

## Methodology

As evident by its name that it should be an ordinary personal dairy but in reality it is not. This program is very useful for many professional persons of instances:

- It is quite useful for doctors. Doctors can use it for taking appointment and also can keep the record of his patients as long as he can.
- The travelers also use it for their purposes, What is their next plan? Where they have to go further? and also they can save their memories of journey etc. for instances.

## Scope of Project

This is a Password Protected Personal Dairy. With this, records for visitors and activities at the organization or institution are absolutely easy. This has a wide array of interesting features like:

- The user can also take a look at these records.
- The user can also make changes in records.
- The user has the power of deleting any record.
- You can delete past records in our dairy.
- You can keep track of the entry and exit times of all visitors including staff.
- You can add records of staff members also into the Dairy.

## Proposed System

Password protected personal dairy is console application without graphics .In this project ,use Can keep their Personal record like they do in a dairy .you can keep record of important things you do in your daily life , like mettings and various other tasks .you can organize your schedule as well as view some particular entry of a particular date or time .In this console application you can add view , edit view and delete records . Records can be added with information such as duration of task , name address, time and date. Storing capacity is unlimited in records(Depends on memory).Its oops concept used on this password protected personal dairy.

## Advantages Over Traditional System

- User can add their daily details safely and it's not time consuming.
- Each section contains high security so there is no chance of data loss.
- There's maximum privacy to the record files.
- Proper validation
- Password protected and Password system in each section.
- High security.
- This system make easy to store daily record .unlike traditional dairy mainting methods which are more tedious, time consuming and prone to error.



## Specific Requirements:

### Data-Flow Diagram

nffhsf

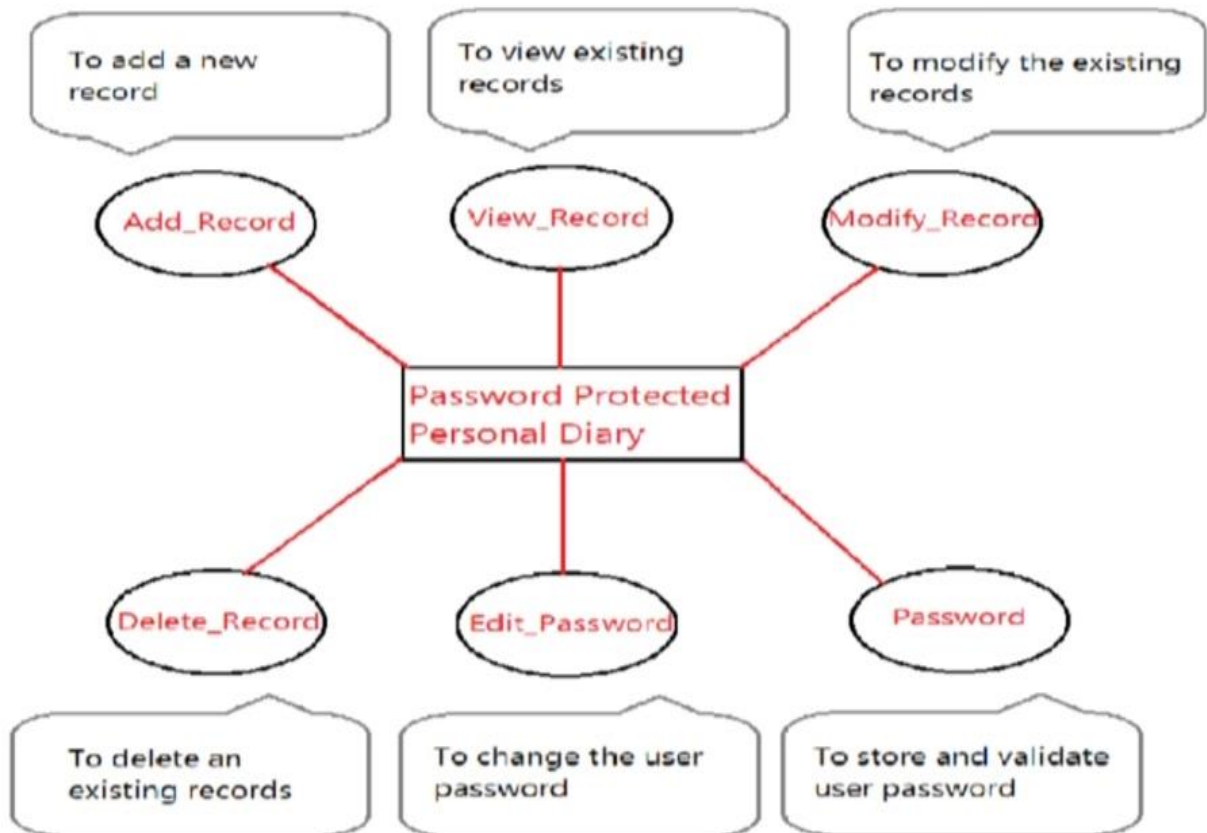


fig.1.1 Data flow Diagram of password protected personal dairy.

## **Description**

### **Hardware Interfaces Requirement**

- Core 2 Duo System.
- 500 Gb HDD.
- 1GB RAM.

### **Software Requirements**

- Windows 7
- Visual Studio 2010.
- C# .Net
- SQL 2008

## Program Coding

```
#include<iostream>
#include<string.h>
#include<stdlib.h>
#include<conio.h>
#include<fstream>
using namespace std;
fstream f,f1,f1;
class record{
    char time[10];
    char name[30];
    char place[25];
    char duration[10];
    char note[500];
    char passwords[25];
public:
    void add_record();
    void view_record();
    void modify_record();
    void delete_record();
    void edit_password();
    int password();
};

void record::add_record()
{
    int i=0;
    system("cls");
    char another = 'Y',time[10];
    record e;
    char filename[15];
    int choice;
int count=0;
    cout<<"\n\n\t\t*****\n";
    cout<<"\t\t* WELCOME TO THE ADD MENU *";
    cout<<"\n\n\t\t*****\n\n";
    cout<<"\n\n\t\tENTER DATE OF YOUR RECORD:[dd-mm-yyyy]:";
    cin.ignore();
    cin.getline(filename,15);
    f.open(filename,ios::in|ios::binary|ios::app|ios::out);
    if(!f.is_open())
    {
        f.open(filename,ios::in|ios::binary|ios::app|ios::out);
        if(!f.is_open())
        {
            cout<<"\nSYSTEM ERROR...";
            cout<<"\nPRESS ANY KEY TO EXIT";
            getch();
            return ;
        }
    }
    while ( another == 'Y' || another=='y' )
    {
        choice=0;
        if(i++>0)
        cin.ignore();
        cout<<"\n\t\tENTER TIME:[hh:mm]:";
        cin.getline(time,10);
        while(f.is_open())
        {
            if(strcmp(e.time,time)==0&&count==0)
            {
                count++;
                cout<<"\n\t\tTHE RECORD ALREADY EXISTS.\n";
                choice=1;
            }
            count=0;
            f.close();
        }
    }
}
```

```
if(choice==0)
{
    f.open(filename,ios::in|ios::binary|ios::app|ios::out);
    strcpy(e.time,time);
    cout<<"\tENTER NAME:";
    cin.getline(e.name,30);
    cout<<"\tENTER PLACE:";
    cin.getline(e.place,25);

    cout<<"\tENTER DURATION:";
    cin.getline(e.duration,10);

    cout<<"\tNOTE:";
    cin.getline(e.note,500);
    f.write((char *)&e,sizeof(e));
    cout<<"\nYOUR RECORD IS ADDED...\n";
}
cout<<"\n\tADD ANOTHER RECORD...(Y/N) ";

another = getchar( ) ;
}
f.close() ;
cout<<"\n\n\tPRESS ANY KEY TO EXIT...";
getch();
}

void record::view_record()
{
    system("cls");
    record customer;
    char time[10],choice,filename[14];
    int ch;
    cout<<"\n\n\t\t*****\n";
    cout<<"\t\t*   HERE IS THE VIEWING MENU   *";
    cout<<"\n\n\t\t*****\n";
    choice=password();
    if(choice!=0)
    {
        return ;
    }
    do
    {
        cout<<"\n\tENTER THE DATE OF RECORD TO BE VIEWED:[dd-mm-yyyy]:";
        cin.ignore();
        cin.getline(filename,15);
        f.open(filename,ios::in|ios::binary);
        if(!f)
        {
            puts ( "\nTHE RECORD DOES NOT EXIST...\n" );
            cout<<"PRESS ANY KEY TO EXIT...";
            getch();
            return ;
        }
        system("cls");
        cout<<"\n\tHOW WOULD YOU LIKE TO VIEW:\n";
        cout<<"\n\t1.WHOLE RECORD OF THE DAY.";
        cout<<"\n\t2.RECORD OF FIX TIME.";
        cout<<"\n\t\tENTER YOUR CHOICE:";
        cin>>ch;
        switch(ch)
        {
            case 1:
                cout<<"\nTHE WHOLE RECORD FOR:"<<filename;
                while (f.read((char *)&customer,sizeof(record)))
                {
                    cout<<"\n";
                    cout<<"\nTIME:"<<customer.time;
                    cout<<"\nMEETING WITH:"<<customer.name;
                    cout<<"\nMEETING AT:"<<customer.place;
                    cout<<"\nDURATION:"<<customer.duration;
                    cout<<"\nNOTE:"<<customer.note;
```

```

        cout<<"\n";
    }
    break;
    case 2:
        cin.ignore();
        cout<<"\nENTER TIME:[hh:mm]:";
        cin.getline(time,10);
        while( f.read((char *)&customer,sizeof(customer)))
        {

            if(strcmp(customer.time,time)==0)
            {
                cout<<"\nYOUR RECORD IS:";
                cout<<"\nTIME:"<<customer.time;
                cout<<"\nMEETING WITH:"<<customer.name;
                cout<<"\nMEETING AT:"<<customer.place;
                cout<<"\nDURATION:"<<customer.duration;
                cout<<"\nNOTE:"<<customer.note;
            }
        }
        break;
    default:
        cout<<"\nYOU TYPED SOMETHING ELSE...\n";
        break;
    }
    cout<<"\n\nWOULD YOU LIKE TO CONTINUE VIEWING...(Y/N):";
    cin>>choice;
    f.close ();
    while(choice=='Y'||choice=='y');
    cout<<"\n\nPRESS ANY KEY TO EXIT...";
    getch();
    return ;
}

void record::modify_record()
{
    system("cls");
    record customer;
    char time[10],choice,filename[14];
    int num,count=0,cnt=0;
    int pos;
    cout<<"\n\n\t\t*****\n";
    cout<<"\t\t* WELCOME TO THE EDITING MENU *";
    cout<<"\n\n\t\t*****\n\n";
    choice=password();
    if(choice!=0)
    {
        return;
    }
    do{
        cout<<"\n\n\tENTER THE DATE OF RECORD TO BE EDITED:[dd-mm-yyyy]:";
        cin.ignore();
        cin.getline(filename,15);
        cout<<"\n\n\tENTER TIME:[hh:mm]:";
        cin.getline(time,10);
        f.open(filename,ios::binary|ios::in);
        if (!f)

        {
            cout<<"\nRECORD DOES NOT EXISTS:";
            cout<<"\nPRESS ANY KEY TO GO BACK";
            getch();
            return;
        }
    }
    while ( f.read((char *)&customer,sizeof(customer)))
    {
        cnt++;
        if(strcmp(customer.time,time)==0)
        {
            cout<<"\nYOUR OLD RECORD WAS AS:";
            cout<<"\nTIME:"<<customer.time;
            cout<<"\nMEETING WITH:"<<customer.name;

```

```
cout<<"\nMEETING AT:"<<customer.place;
cout<<"\nDURATION:"<<customer.duration;
cout<<"\nNOTE:"<<customer.note;
cout<<"\n\n\t\tWHAT WOULD YOU LIKE TO EDIT..";
cout<<"\n1.TIME.";
cout<<"\n2.MEETING PERSON.";
cout<<"\n3.MEETING PLACE.";
cout<<"\n4.DURATION.";
cout<<"\n5.NOTE.";
cout<<"\n6.WHOLE RECORD.";
cout<<"\n7.GO BACK TO MAIN MENU.";
do{
    cout<<"\n\tENTER YOUR CHOICE:";

    cin>>num;

    switch(num)
    {
        case 1:
            cin.ignore();
            cout<<"\nENTER THE NEW DATA:";
            cout<<"\nNEW TIME:[hh:mm]:";
            cin.getline(customer.time,10);
            break;

        case 2:
            cin.ignore();
            cout<<"\nENTER THE NEW DATA:";
            cout<<"\nNEW MEETING PERSON:";
            cin.getline(customer.name,30);
            break;

        case 3:
            cin.ignore();
            cout<<"\nENTER THE NEW DATA:";
            cout<<"\nNEW MEETING PLACE:";
            cin.getline(customer.place,25);
            break;

        case 4:
            cin.ignore();
            cout<<"\nENTER THE NEW DATA:";
            cout<<"\nDURATION:";
            cin.getline(customer.duration,10);
            break;

        case 5:
            cin.ignore();
            cout<<"\nENTER THE NEW DATA:";
            cout<<"\nNOTE:";
            cin.getline(customer.note,500);
            break;

        case 6:
            cin.ignore();
            cout<<"\nENTER THE NEW DATA:";
            cout<<"\nNEW TIME:[hh:mm]:";
            cin.getline(customer.time,10);
            cout<<"\nNEW MEETING PERSON:";
            cin.getline(customer.name,30);
            cout<<"\nNEW MEETING PLACE:";
            cin.getline(customer.place,25);
            cout<<"\nDURATION:";
            cin.getline(customer.duration,10);
            cout<<"\nNOTE:";
            cin.getline(customer.note,500);
            break;

        case 7:
            cin.ignore();
            cout<<"\nPRESS ANY KEY TO GO BACK...\n";
            getch();
            return ;
            break;

        default:
            break;
    }
}
```

```
        cin.ignore();
        cout<<"\nYOU TYPED SOMETHING ELSE...TRY AGAIN\n";
        break;
    }
}while(num<1||num>8);
f.close();
f.open(filename,ios::in|ios::binary|ios::out|ios::ate);
int location=(cnt-1)*sizeof(customer);
f.seekp(location);
f.write((char *)&customer,sizeof(record));
f.close();
f.open(filename,ios::binary|ios::in);
f.seekp(location);
f.read((char *)&customer,sizeof(record));
choice=5;
break;
}
}
if(choice==5)
{
    system("cls");
    cout<<"\n\t\t\tEDITING COMPLETED...\n";
    cout<<"-----\n";
    cout<<"THE NEW RECORD IS:\n";
    cout<<"-----\n";
    cout<<"\nTIME:"<<customer.time;
    cout<<"\nMEETING WITH:"<<customer.name;
    cout<<"\nMEETING AT:"<<customer.place;
    cout<<"\nDURATION:"<<customer.duration;
    cout<<"\nNOTE:"<<customer.note;
    f.close();
    cout<<"\n\n\t\t\tWOULD YOU LIKE TO EDIT ANOTHER RECORD.(Y/N)";
    cin>>choice;
    count++;}
else{
    cout<<"\nTHE RECORD DOES NOT EXIST:\n";
    cout<<"\nWOULD YOU LIKE TO TRY AGAIN...(Y/N)";
    cin>>choice;
    f.close () ;
    cnt=0;
}
}while(choice=='Y'||choice=='y');

if(count==1)
cout<<count<<" FILE IS EDITED...\n";
else if(count>1)
cout<<count<<" FILES ARE EDITED..\n";
else
cout<<"\nNO FILES EDITED...\n";
cout<<"\t\t\tPRESS ENTER TO EXIT EDITING MENU.";
getch();
}
int record::password()
{
    f.open("PASS",ios::in);
    f.read((char*)&passwords,sizeof(passwords));
    f.close();
    char pass[15]={0},check[15]={0},ch;
    int i=0,j;
    cout<<"::FOR SECURITY PURPOSE::";
    cout<<"::ONLY THREE TRIALS ARE ALLOWED::";
    for(j=0;j<3;j++)
    {
        i=0;
        cout<<"\n\n\t\t\tENTER THE PASSWORD:";
        pass[0]=getch();
        while(pass[i]!='\r')
        {
            if(pass[i]=='\b')
            {
                i--;
            }
        }
    }
}
```

```
        cout<<"\b";
        cout<<" ";
        cout<<"\b";
        pass[i]=getch();
    }
    else
    {
        cout<<"*";
        i++;
        pass[i]=getch();
    }
}
pass[i]='\0';
i=0;
if(strcmp(pass,passwords)==0)
{
    cout<<"\n\n\tACCESS GRANTED...\n";
    return 0;
}
else
{
    cout<<"\n\n\tWRONG PASSWORD..\n\n\tACCESS DENIED...\n";
}
}
cout<<"\n\n\t::YOU ENTERED WRONG PASSWORD::YOU ARE NOT ALLOWED TO ACCESS ANY FILE::\n\n\tPRESS ANY KEY TO GO B";
getch();
return 1;
}
void record::edit_password()
{
    system("cls");
    cout<<"\n";
    char pass[15]={0},confirm[15]={0},ch;
    int choice,i,check;
    check=password();
    if(check==1)
    {
        return ;
    }
    do
    {
        if(check==0)
        {
            i=0;
            choice=0;
            cout<<"\n\n\tENTER THE NEW PASSWORD:";
            cin.ignore();
            pass[0]=getch();
            while(pass[i]!='\r')
            {
                if(pass[i]=='\b')
                {
                    i--;
                    cout<<"\b";
                    cout<<" ";
                    cout<<"\b";
                    pass[i]=getch();
                }
                else
                {
                    cout<<"*";
                    i++;
                    pass[i]=getch();
                }
            }
            pass[i]='\0';
            i=0;
            cout<<"\n\n\tCONFIRM PASSWORD:";
            confirm[0]=getch();
            while(confirm[i]!='\r')
            {
                if(confirm[i]=='\b')
```



```

        {
            i--;
            cout<<"\b";
            cout<<" ";
            cout<<"\b";
            confirm[i]=getch();
        }
        else
        {
            cout<<"*";
            i++;
            confirm[i]=getch();
        }
    }
    confirm[i]='\0';
    if(strcmp(pass,confirm)==0)
    {
        strcpy(passwords,pass);
        f.open("PASS",ios::in|ios::out);
        f.seekg(0,ios::beg);
        f.write((char*)&passwords,sizeof(passwords));
        f.close();
        cout<<"\n\n\tPASSWORD CHANGED SUCCESSFULLY.";
        cout<<"\n\n\tPRESS ANY KEY TO CONTINUE...";
        getch();
        return ;
    }
    else
    {
        cout<<"\n\n\tTHE NEW PASSWORD DOES NOT MATCH.";
        choice=1;
    }
}
}while(choice==1);
cout<<"\n\n\tPASSWORD CHANGED...\n\n\tPRESS ANY KEY TO GO BACK...";
getch();
}
void record::delete_record()
{
    system("cls");
    record file;
    char filename[15],another = 'Y',time[10];;
    int choice,check;
    cout<<"\n\n\t\t*****\n";
    cout<<"\n\n\t\t* WELCOME TO DELETE MENU*";
    cout<<"\n\n\t\t*****\n\n";
    check = password();
    if(check==1)
    {
        return ;
    }
    while ( another == 'Y' || another=='y')
    {
        cout<<"\n\n\t\tHOW WOULD YOU LIKE TO DELETE.";
        cout<<"\n\n\t\t#DELETE WHOLE RECORD\t\t\t[1]";
        cout<<"\n\n\t\t#DELETE A PARTICULAR RECORD BY TIME\t\t[2]";
        do
        {
            cout<<"\n\n\t\tENTER YOUR CHOICE:";
            cin>>choice;
            switch(choice)
            {
                case 1:
                    cout<<"\n\n\t\tENTER THE DATE OF RECORD TO BE DELETED:[dd-mm-yyyy].";
                    cin.ignore();
                    cin.getline(filename,15);
                    f.open (filename,ios::in|ios::binary) ;
                    if (!f)
                    {
                        cout<<"\n\n\t\tTHE FILE DOES NOT EXISTS";
                        cout<<"\n\n\t\tPRESS ANY KEY TO GO BACK.";
                        getch();
                    }
                }
            }
        }
    }
}

```

```
        return ;
    }
    f.close();
    remove(filename);
    cout<<"\nDELETED SUCCESFULLY...";
    break;
case 2:
    cout<<"\n\tENTER THE DATE OF RECORD:[dd-mm-yyyy]:";
    cin.ignore();
    cin.getline(filename,15);
    f.open (filename,ios::in|ios::binary) ;
    if (!f)
    {
        cout<<"\nTHE FILE DOES NOT EXISTS";
        cout<<"\nPRESS ANY KEY TO GO BACK.";
        getch();
        return ;
    }
    fi.open("temp",ios::in|ios::out|ios::binary|ios::app|ios::ate);
    if(!fi)
    {
        cout<<"\nSYSTEM ERROR";
        cout<<"\nPRESS ANY KEY TO GO BACK";
        getch();
        return ;
    }
    cout<<"\n\tENTER THE TIME OF RECORD TO BE DELETED:[hh:mm]:";
    cin.getline(time,10);
    while(f.read((char*)&file,sizeof(file)))
    {
        if(strcmp(file.time,time)!=0)
            fi.write((char*)&file,sizeof(file));
    }
    f.close();
    fi.close();
    remove(filename);
    rename("temp",filename);
    cout<<"\nDELETED SUCCESFULLY...";
    break;
default:
    cout<<"\n\tYOU ENTERED WRONG CHOICE";
    break;
}
}while(choice<1||choice>2);
cout<<"\n\tDO YOU LIKE TO DELETE ANOTHER RECORD.(Y/N):";
    cin>>another;
}
cout<<"\n\n\tPRESS ANY KEY TO EXIT...";
getch();
}

int main()
{
    int ch;
    record d;
    cout<<"\n\n\t*****\n";
    cout<<"\t*PASSWORD PROTECTED PERSONAL DIARY*\n";
    cout<<"\t*****";
    while(1)
    {
        cout<<"\n\n\t\tMAIN MENU:";
        cout<<"\n\t\t-----\n";
        cout<<"\n\n\t\tADD RECORD\t[1]";
        cout<<"\n\n\t\tVIEW RECORD\t[2]";
        cout<<"\n\n\t\tMODIFY RECORD\t[3]";
        cout<<"\n\n\t\tDELETE RECORD\t[4]";
        cout<<"\n\n\t\tEDIT PASSWORD\t[5]";
        cout<<"\n\n\t\tEXIT\t[6]";
        cout<<"\n\n\t\tENTER YOUR CHOICE:";

        cin>>ch;
        switch(ch)
```

```
{
    case 1:
        d.add_record();
        break;
    case 2:
        d.view_record();
        break;
    case 3:
        d.modify_record();
        break;
    case 4:
        d.delete_record();
        break;
    case 5:
        d.edit_password();
        break;
    case 6:
        cout<<"\n\n\t\tTHANK YOU!!!";
        getch();
        exit(0);
    default:
        cout<<"\nYOU ENTERED WRONG CHOICE..";
        cout<<"\nPRESS ANY KEY TO TRY AGAIN";
        getch();
        break;
}
    system("cls");
}
return 0;
}
```

**OUTPUT Screen:**

**EXIT**



**Fig.1.2 OUTPUT OF PROGRAM**

## Conclusion

- Application was deployed successfully in device & it give good result as per expectation.
- Password protected personal dairy very useful to professional person for any instance.