# Personal Diary Management System

It’s a convenient software for daily use. By using this software, you can easily manage your daily schedule, short notes, and all your client information. All information will store in a file using encryption so that, you don’t need to worry about your privacy. Without your consent, no one will be able access your personal data. It’s easy to use because of its friendly and clean interface.

Contents

[Personal Diary Management System 1](#_Toc101871645)

[**How to run:** 2](#_Toc101871646)

[**Key Features** 2](#_Toc101871647)

[1. Notebooks 2](#_Toc101871648)

[a. Add New Notes 2](#_Toc101871649)

[b. View Notes 3](#_Toc101871650)

[c. Edit Notes 3](#_Toc101871651)

[d. Delete Notes 4](#_Toc101871652)

[2. Daily Schedule 4](#_Toc101871653)

[a. Add New Schedule 4](#_Toc101871654)

[b. View Schedule 5](#_Toc101871655)

[c. Edit Schedule 5](#_Toc101871656)

[d. Delete Schedule 5](#_Toc101871657)

[3. Client Information 6](#_Toc101871658)

[a. Add New Client 6](#_Toc101871659)

[b. View Client Information 7](#_Toc101871660)

[c. Edit Client Information 7](#_Toc101871661)

[d. Delete Client Information 7](#_Toc101871662)

[4. Security Settings 8](#_Toc101871663)

[a. Update Password 8](#_Toc101871664)

[b. Update Security Question 8](#_Toc101871665)

[5. About Software 9](#_Toc101871666)

[6. UI Design 9](#_Toc101871667)

[a. Welcome Screen 9](#_Toc101871668)

[b. Header Screen 9](#_Toc101871669)

[c. Exit Screen 10](#_Toc101871670)

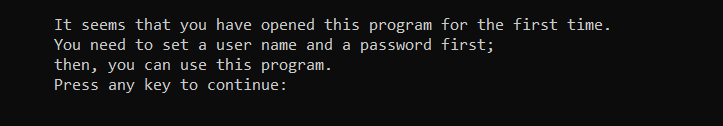
[Usage C Standard Function 10](#_Toc101871671)

[Source Code: 11](#_Toc101871672)

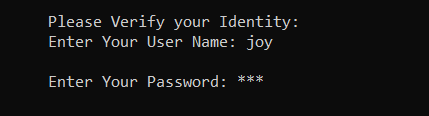
[Team Info.: 76](#_Toc101871673)

# **How to run:**

If user open this program first time, program will ask to set new username, password and security questions.



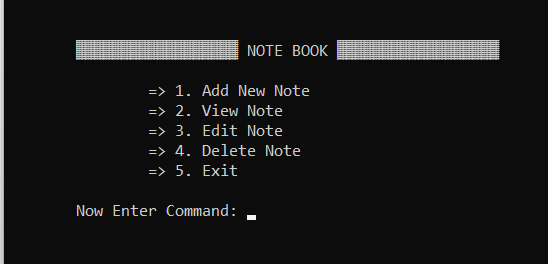
If user open this program next time, it will ask user to verify username and password.



In verification, password will not be shown.

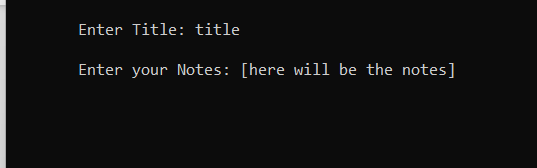
# **Key Features**

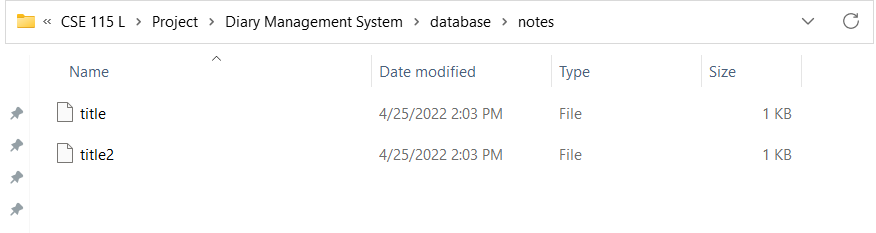
## Notebooks

This function is for managing your personal notes. Here we have four options. The first one is for adding new notes, second one is for viewing added notes, third one is for editing existing notes, and the last one for deleting previously added notes. 

### Add New Notes

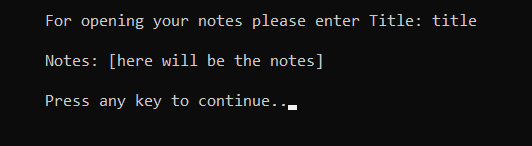
In this option, user will input two data. One for title and another one for notes. The notes will be saved in a file. The title will be used as a file name and the notes will be stored in this file with encryption and all the notes’ file will be stored in a single specific folder.





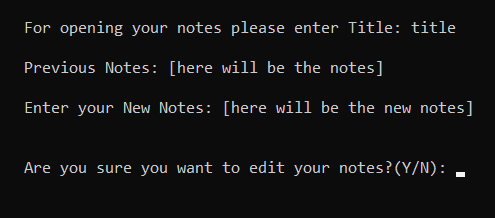
### View Notes

Here, user just needs to enter the title of the notes which was previously added. If the given title is available in database, then the console will show the notes by decrypting stored data.



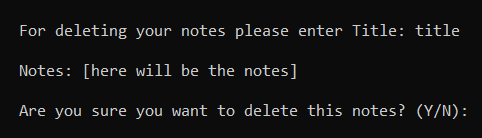
### Edit Notes

It’s an interesting feature that can edit user’s previously added notes. First, it will display the existing notes that, user want to edit. Then, it will ask for new data that, user want to change. After that console will ask for a confirmation and an identity verification will be displayed.



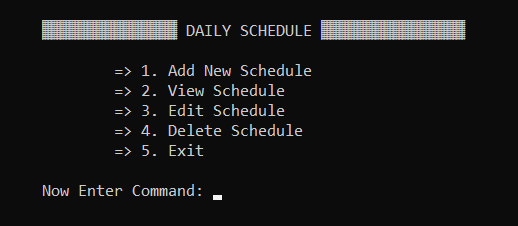
### Delete Notes

This option is specifically used for deleting previously added notes. User will be asked for the title only. Console will show the notes and ask for a confirmation.



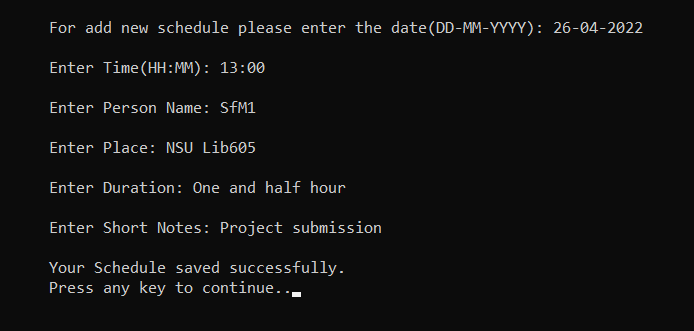
## Daily Schedule

Daily schedule is used for user’s daily time management necessities. Its features are given below.



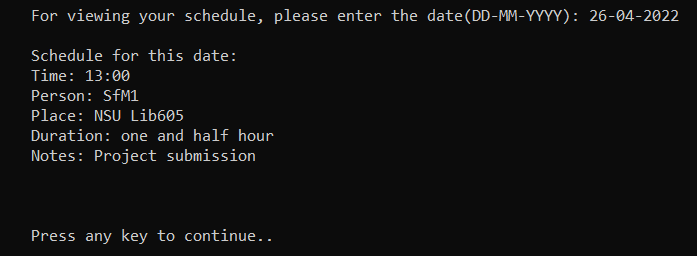
### Add New Schedule

In this feature, user can easily add daily time schedule for meetings, works etc. User will be asked to enter date, time, person name, place, duration and short notes. Schedule will be stored in a file named by the dates with encryption.



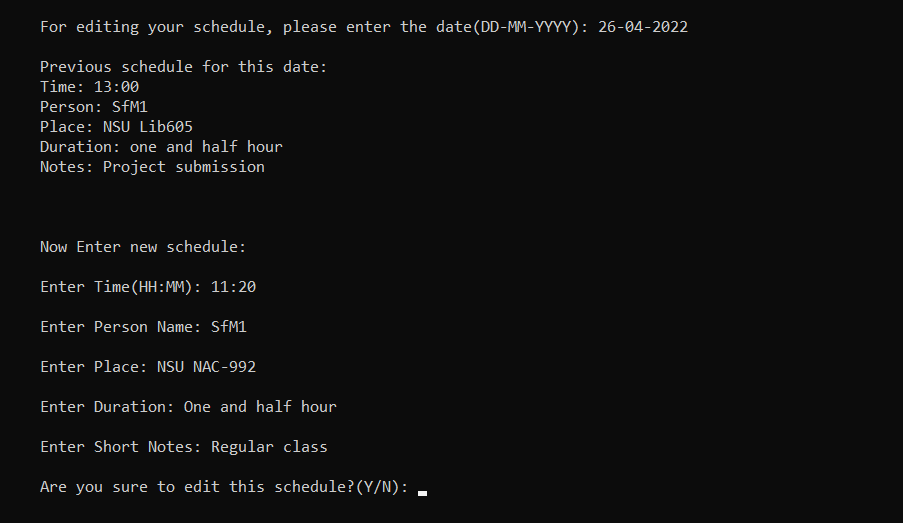
### View Schedule

This feature will display all the schedule according to dates. It will ask user to enter the date only.



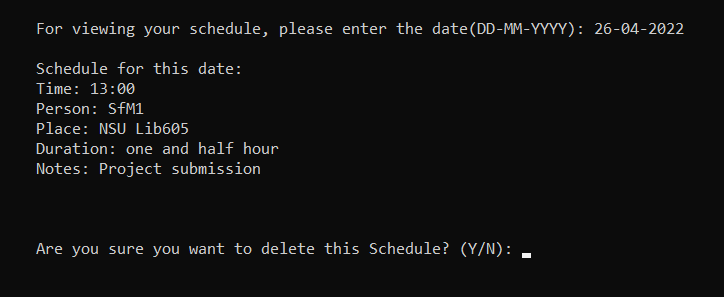
### Edit Schedule

Edit Schedule is used for editing existing time schedules which are added by the user previously. User need to enter whole data again. Then, console will ask for the confirmation.



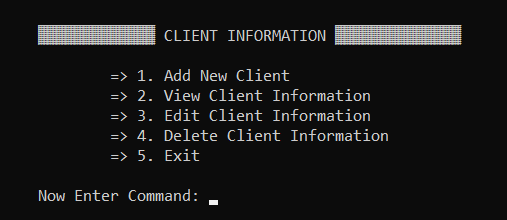
### Delete Schedule

This option is used for deleting user’s previous schedule by date. Console will ask for confirmation and identity verification.



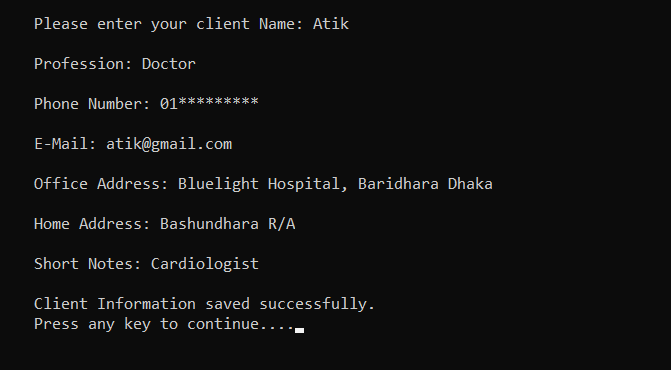
## Client Information

This is the most detailed feature in this whole software. It contains the features given below.



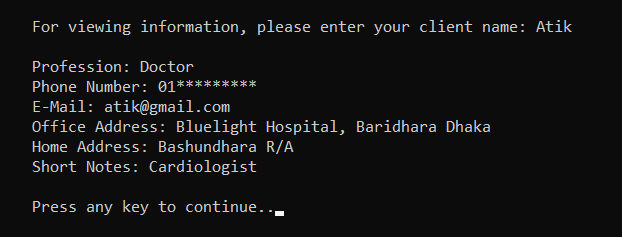
### Add New Client

This feature gives user ability to add new clients by their names. Also, adds details such as, profession, phone number, E-mail, office address, home address and short notes. All the information will be sorted by their names with encryption.



### View Client Information

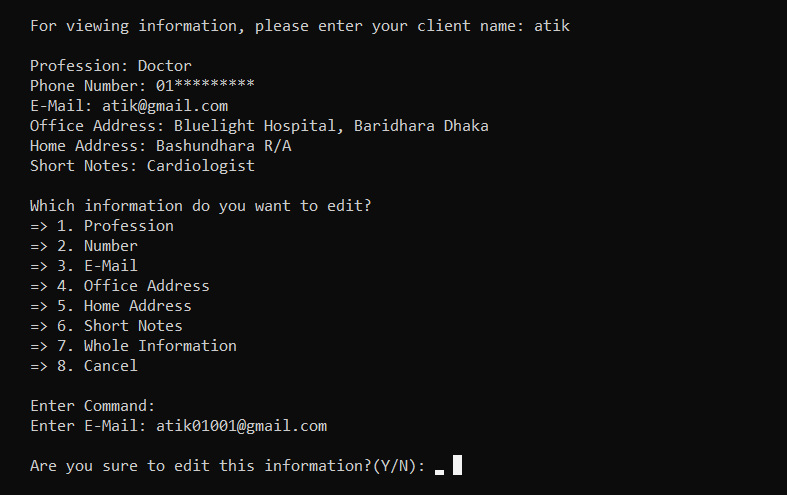
In this feature console will display client details according to client names.



### Edit Client Information

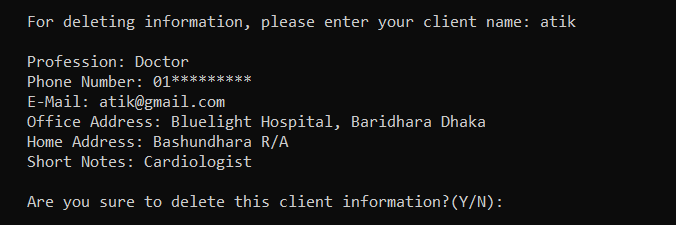
If user want to edit any client information, this option will help to do that. User can edit a specific field or the whole information by some simple commands.

At the end, console will ask for confirmation and identity verification.



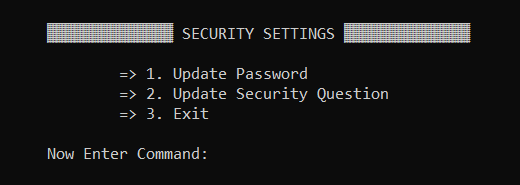
### Delete Client Information

This option is useful for deleting existing client information. Console will ask user for confirmation and verification after entering the client’s name.



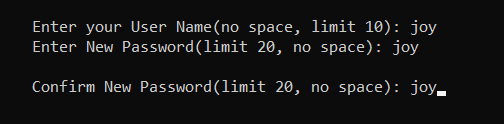
## Security Settings

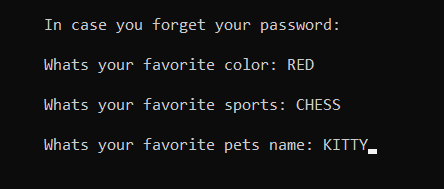
Here, user can manage security settings such as, change password and security question.



### Update Password

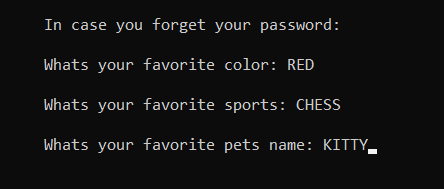
This option is used for changing the old password with the new one. First console will ask for the identity verification. Then, it will ask for new password and security questions.





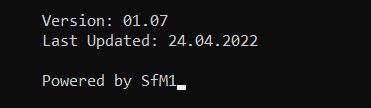
### Update Security Question

Here, console will ask user to verify identity. Then it will ask for new security question.



## About Software

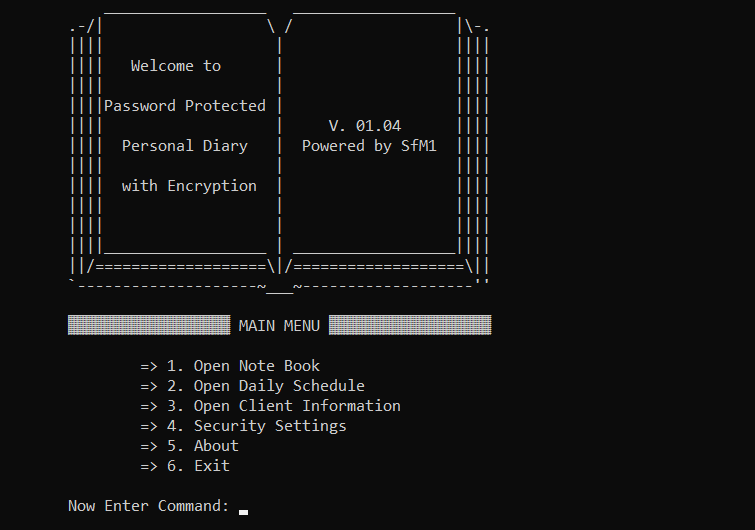
It contains the version details, last update date and credits.



## UI Design

We added some interesting asci arts which are given below.

### Welcome Screen



### Header Screen

Header Screen

### Exit Screen



# Usage C Standard Function

|  |  |  |
| --- | --- | --- |
| Function | Header File | Brief Description |
| fclose | stdio.h | Closes the specified stream. |
| fgetc | stdio.h | Reads a single unsigned character from the input stream. |
| fgets | stdio.h | Reads a string from the input stream. |
| fopen | stdio.h | Opens the specified file. |
| fprintf | stdio.h | Formats and prints characters and values to the output stream. |
| gets | stdio.h | Reads a string from stdin and stores it in buffer. |
| printf | stdio.h | Formats and prints characters and values to stdout. |
| remove | stdio.h | Deletes the file specified by filename. |
| scanf | stdio.h | Reads data from stdin into locations given by arg-list. |
| strcat | string.h | Concatenates string2 to string1. |
| strcmp | string.h | Compares the value of string1 to string2. |
| strcpy | string.h | Copies string2 into string1. |
| strlen | string.h | Calculates the length of string. |
| system | stdlib.h | Passes string to the system command analyzer. |
| getwcd | unistd.h | Copy current working directory into a string |
| mkdir | unistd.h | Creates folder by folder direction or folder name |
| getch | conio.h | Reads a single unsigned character from the user |

# Source Code:

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <conio.h>

#include <unistd.h>

void CHECK\_DIRECTORY(void);

void DIRECTORY\_CREATE(void);

void SET\_PASSWORD(void);

void IDENTITY\_VERIFY(void);

void UPDATE\_PASSWORD(void);

void UPDATE\_SECURITY\_QUESTION(void);

void RESET\_PASSWORD(void);

void ADD\_NOTES(void);

void VIEW\_NOTES(void);

void EDIT\_NOTES(void);

void DELETE\_NOTES(void);

void ADD\_SCHEDULE(void);

void VIEW\_SCHEDULE(void);

void EDIT\_SCHEDULE(void);

void DELETE\_SCHEDULE(void);

void ADD\_CLIENT\_INFORMATION(void);

void VIEW\_CLIENT\_INFORMATION(void);

void EDIT\_CLIENT\_INFORMATION(void);

void DELETE\_CLIENT\_INFORMATION(void);

void ABOUT(void);

void WELCOME\_SCREEN(void);

void HEADER\_SCREEN(char menu\_msg[]);

void EXIT\_SCREEN(void);

//global variable

char current\_directory[1000];

char admin\_directory[1000];

char filename\_admin[] = "\\database\\admin";

char reverse\_current\_directory[1000];

char folder\_notes[] = "/database/notes";

char folder\_client\_information[] = "/database/client\_information";

char folder\_daily\_schedule[] = "/database/daily\_schedule";

char notes\_directory[1000];

char client\_information\_directory[1000];

char daily\_schedule\_directory[1000];

int main()

{

char command;

CHECK\_DIRECTORY();

system("cls");

IDENTITY\_VERIFY();

while(1){

system("cls");

WELCOME\_SCREEN();

HEADER\_SCREEN("MAIN MENU");

printf("\n\t\t=> 1. Open Note Book\n\t\t=> 2. Open Daily Schedule\n\t\t=> 3. Open Client Information\n\t\t=> 4. Security Settings\n\t\t=> 5. About\n\t\t=> 6. Exit\n\n\tNow Enter Command: ");

command = getch();

if(command == '1'){

while(1){

system("cls");

HEADER\_SCREEN("NOTE BOOK");

printf("\n\t\t=> 1. Add New Note\n\t\t=> 2. View Note\n\t\t=> 3. Edit Note\n\t\t=> 4. Delete Note\n\t\t=> 5. Exit\n\n\tNow Enter Command: ");

command = getch();

if(command == '1'){

ADD\_NOTES();

}

else if(command == '2'){

VIEW\_NOTES();

}

else if(command == '3'){

EDIT\_NOTES();

}

else if(command == '4'){

DELETE\_NOTES();

}

else if(command == '5'){

break;

}

else{

printf("\n\tWrong Command!! Press any key to try again..");

getch();

continue;

}

}

}

else if(command == '2'){

while(1){

system("cls");

HEADER\_SCREEN("DAILY SCHEDULE");

printf("\n\t\t=> 1. Add New Schedule\n\t\t=> 2. View Schedule\n\t\t=> 3. Edit Schedule\n\t\t=> 4. Delete Schedule\n\t\t=> 5. Exit\n\n\tNow Enter Command: ");

command = getch();

if(command == '1'){

ADD\_SCHEDULE();

}

else if(command == '2'){

VIEW\_SCHEDULE();

}

else if(command == '3'){

EDIT\_SCHEDULE();

}

else if(command == '4'){

DELETE\_SCHEDULE();

}

else if(command == '5'){

break;

}

else{

printf("\n\tWrong Command!! Press any key to try again..");

getch();

continue;

}

}

}

else if(command == '3'){

while(1){

system("cls");

HEADER\_SCREEN("CLIENT INFORMATION");

printf("\n\t\t=> 1. Add New Client\n\t\t=> 2. View Client Information\n\t\t=> 3. Edit Client Information\n\t\t=> 4. Delete Client Information\n\t\t=> 5. Exit\n\n\tNow Enter Command: ");

command = getch();

if(command == '1'){

ADD\_CLIENT\_INFORMATION();

}

else if(command == '2'){

VIEW\_CLIENT\_INFORMATION();

}

else if(command == '3'){

EDIT\_CLIENT\_INFORMATION();

}

else if(command == '4'){

DELETE\_CLIENT\_INFORMATION();

}

else if(command == '5'){

break;

}

else{

printf("\n\tWrong Command!! Press any key to try again..");

getch();

continue;

}

}

}

else if(command == '4'){

while(1){

system("cls");

HEADER\_SCREEN("SECURITY SETTINGS");

printf("\n\t\t=> 1. Update Password\n\t\t=> 2. Update Security Question\n\t\t=> 3. Exit\n\n\tNow Enter Command: ");

command = getch();

if(command == '1'){

UPDATE\_PASSWORD();

}

else if(command == '2'){

UPDATE\_SECURITY\_QUESTION();

}

else if(command == '3'){

break;

}

else{

printf("\n\tWrong Command!! Press any key to try again..");

getch();

continue;

}

}

}

else if(command == '5'){

ABOUT();

}

else if(command == '6'){

EXIT\_SCREEN();

break;

}

else{

printf("\n\tWrong Command!! Press any key to try again..");

getch();

continue;

}

}

}

void CHECK\_DIRECTORY(void)

{

FILE \*check\_directory;

int i, j, length\_current\_directory, length\_filename\_admin;

getcwd(current\_directory, 1000);

length\_current\_directory = strlen(current\_directory);

length\_filename\_admin = strlen(filename\_admin);

strcpy(admin\_directory, current\_directory);

strcat(admin\_directory, filename\_admin);

check\_directory = fopen(admin\_directory, "r");

if(check\_directory == NULL){

printf("\n\tIt seems that you have opened this program for the first time.\n\tYou need to set a user name and a password first;\n\tthen, you can use this program.\n\tPress any key to continue: ");

getch();

DIRECTORY\_CREATE();

SET\_PASSWORD();

}

else{

fclose(check\_directory);

}

}

void DIRECTORY\_CREATE(void)

{

int i, j, temp;

FILE \*create\_file\_admin;

for( i = 0; i < strlen(current\_directory); i++){

if(current\_directory[i] == '\\'){

reverse\_current\_directory[i] = '/';

}

else{

reverse\_current\_directory[i] = current\_directory[i];

}

}

reverse\_current\_directory[i] = '\0';

strcpy(notes\_directory, reverse\_current\_directory);

strcpy(client\_information\_directory, reverse\_current\_directory);

strcpy(daily\_schedule\_directory, reverse\_current\_directory);

strcat(notes\_directory, folder\_notes);

strcat(client\_information\_directory, folder\_client\_information);

strcat(daily\_schedule\_directory, folder\_daily\_schedule);

mkdir("database");

mkdir(notes\_directory);

mkdir(client\_information\_directory);

mkdir(daily\_schedule\_directory);

create\_file\_admin = fopen(admin\_directory, "w");

fclose(create\_file\_admin);

}

void SET\_PASSWORD(void)

{

FILE \*open\_admin;

char user\_name[10];

char password[20];

char password\_confirm[20];

char color[20];

char sports[20];

char pets[20];

int i;

open\_admin = fopen(admin\_directory, "w");

if(open\_admin == NULL){

printf("\n\tSystem File Missing..");

}

else{

system("cls");

printf("\n\tEnter your new User Name(no space, limit 10): ");

scanf("%s", user\_name);

getchar();

for(i = 0; i < strlen(user\_name); i++){

fprintf(open\_admin, "%c", user\_name[i] + 508);

}

fprintf(open\_admin, "\n");

while(1){

system("cls");

printf("\n\tEnter your User Name(no space, limit 10): %s", user\_name);

printf("\n\tEnter New Password(limit 20, no space): ");

scanf("%s", password);

getchar();

printf("\n\tConfirm New Password(limit 20, no space): ");

scanf("%s", password\_confirm);

getchar();

if(strcmp(password, password\_confirm) == 0){

for(i = 0; i < strlen(password); i++){

fprintf(open\_admin, "%c", password[i] + 508);

}

break;

}

else{

printf("\n\tPassword doesn't match. Press any key to try again: ");

getch();

continue;

}

}

system("cls");

printf("\n\tPassword set successfully.\n\tPress any key to continue: ");

getch();

fprintf(open\_admin, "\n");

system("cls");

printf("\n\tIn case you forget your password:\n\n\tWhats your favorite color: ");

scanf("%s", color);

getchar();

printf("\n\tWhats your favorite sports: ");

scanf("%s", sports);

getchar();

printf("\n\tWhats your favorite pets name: ");

scanf("%s", pets);

getchar();

for(i = 0; i < strlen(color); i++){

fprintf(open\_admin, "%c", color[i] + 508);

}

fprintf(open\_admin, "\n");

for(i = 0; i < strlen(sports); i++){

fprintf(open\_admin, "%c", sports[i] + 508);

}

fprintf(open\_admin, "\n");

for(i = 0; i < strlen(pets); i++){

fprintf(open\_admin, "%c", pets[i] + 508);

}

system("cls");

printf("\n\tSecurity Question set successfully.\n\tPress any key to continue: ");

getch();

fclose(open\_admin);

}

}

void IDENTITY\_VERIFY(void)

{

FILE \*open\_admin;

char user\_name[10];

char user\_name\_input[10];

char password[20];

char password\_input[20];

int i, run = 1, access = 0;

char command, ch = ' ';

open\_admin = fopen(admin\_directory, "r");

if(open\_admin == NULL){

printf("\n\tSystem File Missing..Press any key to continue..");

getch();

}

else{

fgets(user\_name, 10, open\_admin);

for(i = 0; i < strlen(user\_name); i++){

user\_name[i] = user\_name[i] - 508;

}

user\_name[i-1] = '\0';

fgets(password, 20, open\_admin);

for(i = 0; i < strlen(password); i++){

password[i] = password[i] - 508;

}

password[strlen(password)-1] = '\0';

while(run < 4){

system("cls");

printf("\n\tPlease Verify your Identity:");

printf("\n\tEnter Your User Name: ");

scanf("%s", user\_name\_input);

getchar();

if(strcmp(user\_name, user\_name\_input) == 0){

printf("\n\tEnter Your Password: ");

for(i = 0; i < 20; i++){

password\_input[i] = getch();

ch = password\_input[i];

if(ch == 13){

break;

}

else{

printf("\*");

}

}

password\_input[i] = '\0';

if(strcmp(password, password\_input) == 0){

printf("\n\n\tACCESS GRANTED..\n\n\tPress Any Key to Continue..");

getch();

system("cls");

access = 1;

break;

}

else{

printf("\n\tWrong Password. Press any key to try Again..");

getch();

system("cls");

run++;

continue;

}

}

else{

printf("\n\tWrong User Name. Press any key to try Again..");

getch();

system("cls");

continue;

}

}

fclose(open\_admin);

if(access == 0){

while(1){

system("cls");

printf("\n\tIt seems that you entered the wrong cardinals three times.\n\tDo you forget your password?\n\tWant to reset it?(Y/N)\n\n\tEnter Command: ");

command = getch();

if(command == 'Y' || command == 'y'){

RESET\_PASSWORD();

break;

}

else{

printf("\n\tWrong Command!! Press any key to try again..");

getch();

continue;

}

}

}

}

}

void UPDATE\_PASSWORD(void)

{

IDENTITY\_VERIFY();

SET\_PASSWORD();

}

void UPDATE\_SECURITY\_QUESTION(void)

{

FILE \*open\_admin;

char user\_name[20];

char password[20];

char color[20];

char sports[20];

char pets[20];

int i;

IDENTITY\_VERIFY();

open\_admin = fopen(admin\_directory, "r");

if(open\_admin == NULL){

printf("\n\tSystem File Missing...\n\tPress any key to continue..");

getch();

}

else{

fgets(user\_name, 20, open\_admin);

for(i = 0; i < strlen(user\_name); i++){

user\_name[i] = user\_name[i] - 508;

}

user\_name[i-1] = '\0';

fgets(password, 20, open\_admin);

for(i = 0; i < strlen(password); i++){

password[i] = password[i] - 508;

}

password[i-1] = '\0';

fclose(open\_admin);

system("cls");

printf("\n\tIn case you forget your password:\n\n\tWhats your favorite color: ");

scanf("%s", color);

getchar();

printf("\n\tWhats your favorite sports: ");

scanf("%s", sports);

getchar();

printf("\n\tWhats your favorite pets name: ");

scanf("%s", pets);

getchar();

open\_admin = fopen(admin\_directory, "w");

if(open\_admin == NULL){

printf("\n\tSecurity Question Update Failed.. Press any key to continue..");

getch();

}

else{

for(i = 0; i < strlen(user\_name); i++){

fprintf(open\_admin, "%c", user\_name[i] + 508);

}

fprintf(open\_admin, "\n");

for(i = 0; i < strlen(password); i++){

fprintf(open\_admin, "%c", password[i] + 508);

}

fprintf(open\_admin, "\n");

for(i = 0; i < strlen(color); i++){

fprintf(open\_admin, "%c", color[i] + 508);

}

fprintf(open\_admin, "\n");

for(i = 0; i < strlen(sports); i++){

fprintf(open\_admin, "%c", sports[i] + 508);

}

fprintf(open\_admin, "\n");

for(i = 0; i < strlen(pets); i++){

fprintf(open\_admin, "%c", pets[i] + 508);

}

system("cls");

printf("\n\tSecurity Question Updated Successfully.\n\tPress any key to continue..");

getch();

fclose(open\_admin);

}

}

}

void RESET\_PASSWORD(void)

{

FILE \*open\_admin;

char color[20];

char sports[20];

char pets[20];

char color\_input[20];

char sports\_input[20];

char pets\_input[20];

char temp\_string[20];

int i;

open\_admin = fopen(admin\_directory, "r");

if(open\_admin == NULL){

printf("\n\tSystem File Missing.. Press any key to continue...");

getch();

}

else{

fgets(temp\_string, 20, open\_admin);

fgets(temp\_string, 20, open\_admin);

fgets(color, 20, open\_admin);

for(i = 0; i < strlen(color); i++){

color[i] = color[i] - 508;

}

color[i-1] = '\0';

fgets(sports, 20, open\_admin);

for(i = 0; i < strlen(sports); i++){

sports[i] = sports[i] - 508;

}

sports[i-1] = '\0';

fgets(pets, 20, open\_admin);

for(i = 0; i < strlen(pets); i++){

pets[i] = pets[i] - 508;

}

while(1){

system("cls");

printf("\n\tWhats your favorite color: ");

gets(color\_input);

if(strcmp(color, color\_input) == 0){

printf("\n\tWhats your favorite sports: ");

gets(sports\_input);

if(strcmp(sports, sports\_input) == 0){

printf("\n\tWhats your favorite pets name: ");

gets(pets\_input);

if(strcmp(pets, pets\_input) == 0){

printf("\n\tAccess Granted..\n\tPress any key to continue..");

getch();

fclose(open\_admin);

SET\_PASSWORD();

break;

}

else{

printf("\n\tWrong Input.. Press any key to try again..");

getch();

continue;

}

}

else{

printf("\n\tWrong Input.. Press any key to try again..");

getch();

continue;

}

}

else{

printf("\n\tWrong Input.. Press any key to try again..");

getch();

continue;

}

}

}

}

void ADD\_NOTES(void)

{

system("cls");

char title[100];

char notes[1000];

char notes\_direction\_file[1000];

FILE \*open\_notes\_file;

while(1){

system("cls");

printf("\n\tEnter Title: ");

gets(title);

strcpy(notes\_direction\_file, current\_directory);

strcat(notes\_direction\_file, "\\database\\notes\\");

strcat(notes\_direction\_file, title);

open\_notes\_file = fopen(notes\_direction\_file, "r");

if(open\_notes\_file != NULL){

printf("\n\tIt seems that you already have a notes by this name.\n\tYou can't add two notes by the same name.\n\tPress any key to try again...");

getch();

fclose(open\_notes\_file);

continue;

}

else{

break;

}

}

open\_notes\_file = fopen(notes\_direction\_file, "w");

if(open\_notes\_file == NULL){

printf("\n\tUnable to create file..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tEnter your Notes: ");

gets(notes);

for(int i = 0; i < strlen(notes); i++){

fprintf(open\_notes\_file, "%c", notes[i] + 508);

}

printf("\n\tYour notes saved successfully.\n\tPress any key to continue..");

getch();

fclose(open\_notes\_file);

}

}

void VIEW\_NOTES(void)

{

system("cls");

char title[100];

char ch;

printf("\n\tFor opening your notes please enter Title: ");

gets(title);

char notes\_direction\_file[1000];

strcpy(notes\_direction\_file, current\_directory);

strcat(notes\_direction\_file, "\\database\\notes\\");

strcat(notes\_direction\_file, title);

FILE \*open\_notes\_file;

open\_notes\_file = fopen(notes\_direction\_file, "r");

if(open\_notes\_file == NULL){

printf("\n\tNotes not found in database..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tNotes: ");

while(1){

ch = fgetc(open\_notes\_file);

if(ch == EOF){

break;

}

else{

printf("%c", ch-508);

}

}

printf("\n\n\tPress any key to continue..");

getch();

fclose(open\_notes\_file);

}

}

void EDIT\_NOTES(void)

{

FILE \*open\_notes\_file;

char title[100];

char notes[1000];

char ch;

system("cls");

printf("\n\tFor opening your notes please enter Title: ");

gets(title);

char notes\_direction\_file[1000];

strcpy(notes\_direction\_file, current\_directory);

strcat(notes\_direction\_file, "\\database\\notes\\");

strcat(notes\_direction\_file, title);

open\_notes\_file = fopen(notes\_direction\_file, "r");

if(open\_notes\_file == NULL){

printf("\n\tNotes not found in database..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tPrevious Notes: ");

while(1){

ch = fgetc(open\_notes\_file);

if(ch == EOF){

break;

}

else{

printf("%c", ch-508);

}

}

fclose(open\_notes\_file);

printf("\n\n\tEnter your New Notes: ");

gets(notes);

printf("\n\n\tAre you sure you want to edit your notes?(Y/N): ");

ch = getch();

if(ch == 'Y' || ch == 'y'){

IDENTITY\_VERIFY();

open\_notes\_file = fopen(notes\_direction\_file, "w");

if(open\_notes\_file == NULL){

printf("\n\tUnable to Edit Notes..\n\tPress any key to continue..");

getch();

}

else{

for(int i = 0; i < strlen(notes); i++){

fprintf(open\_notes\_file, "%c", notes[i] + 508);

}

printf("\n\n\tYour notes edited successfully.\n\tPress any key to continue..");

getch();

fclose(open\_notes\_file);

}

}

else{

printf("\n\n\tEdit canceled..\n\tPress any key to Continue...");

getch();

}

}

}

void DELETE\_NOTES(void)

{

FILE \*open\_notes\_file;

char title[100];

char ch;

int status;

system("cls");

printf("\n\tFor deleting your notes please enter Title: ");

gets(title);

char notes\_direction\_file[1000];

strcpy(notes\_direction\_file, current\_directory);

strcat(notes\_direction\_file, "\\database\\notes\\");

strcat(notes\_direction\_file, title);

open\_notes\_file = fopen(notes\_direction\_file, "r");

if(open\_notes\_file == NULL){

printf("\n\tNotes not found in database..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tNotes: ");

while(1){

ch = fgetc(open\_notes\_file);

if(ch == EOF){

break;

}

else{

printf("%c", ch-508);

}

}

fclose(open\_notes\_file);

printf("\n\n\tAre you sure you want to delete this notes? (Y/N): ");

ch = getch();

if(ch == 'Y' || ch == 'y'){

IDENTITY\_VERIFY();

status = remove(notes\_direction\_file);

if(status == 0){

printf("\n\tFile Deleted successfully..\n\tPress any key to continue...");

getch();

}

else{

printf("\n\tFile Delete failed..\n\tPress any key to try again...");

getch();

}

}

else{

printf("\n\tDelete canceled...\n\tPress any key to continue..");

getch();

}

}

}

void ADD\_SCHEDULE(void)

{

FILE \*open\_schedule\_file;

char date[200];

char time[200];

char person[200];

char place[200];

char duration[200];

char notes[1000];

char schedule\_file\_direction[1000];

char ch;

char variable\_name\_Time[] = "\tTime: ";

char variable\_name\_Person[] = "\n\tPerson: ";

char variable\_name\_Place[] = "\n\tPlace: ";

char variable\_name\_Duration[] = "\n\tDuration: ";

char variable\_name\_Notes[] = "\n\tNotes: ";

char variable\_name\_New\_Line[] = "\n\n";

int i;

system("cls");

printf("\n\tFor add new schedule please enter the date(DD-MM-YYYY): ");

gets(date);

strcpy(schedule\_file\_direction, current\_directory);

strcat(schedule\_file\_direction, "\\database\\daily\_schedule\\");

strcat(schedule\_file\_direction, date);

open\_schedule\_file = fopen(schedule\_file\_direction, "r");

if(open\_schedule\_file == NULL){

open\_schedule\_file = fopen(schedule\_file\_direction, "w");

if(open\_schedule\_file == NULL){

printf("\n\tUnable to create file..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tEnter Time(HH:MM): ");

gets(time);

printf("\n\tEnter Person Name: ");

gets(person);

printf("\n\tEnter Place: ");

gets(place);

printf("\n\tEnter Duration: ");

gets(duration);

printf("\n\tEnter Short Notes: ");

gets(notes);

for(i = 0; i < strlen(variable\_name\_Time); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Time[i]+508);

}

for(i = 0; i < strlen(time); i++){

fprintf(open\_schedule\_file, "%c", time[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Person); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Person[i]+508);

}

for(i = 0; i < strlen(person); i++){

fprintf(open\_schedule\_file, "%c", person[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Place); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Place[i]+508);

}

for(i = 0; i < strlen(place); i++){

fprintf(open\_schedule\_file, "%c", place[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Duration); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Duration[i]+508);

}

for(i = 0; i < strlen(duration); i++){

fprintf(open\_schedule\_file, "%c", duration[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Notes); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Notes[i]+508);

}

for(i = 0; i < strlen(notes); i++){

fprintf(open\_schedule\_file, "%c", notes[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_New\_Line); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_New\_Line[i]+508);

}

printf("\n\tYour Schedule saved successfully.\n\tPress any key to continue..");

getch();

fclose(open\_schedule\_file);

}

}

else{

fclose(open\_schedule\_file);

printf("\n\tIt seems that you already have a schedule for this date.\n\tDo you want to add another plan at a different time,\n\tor you can cancel and add another program on a different date?(Y/N): ");

ch = getch();

if(ch == 'Y' || ch == 'y'){

open\_schedule\_file = fopen(schedule\_file\_direction, "a");

if(open\_schedule\_file == NULL){

printf("\n\tUnable to open file..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tEnter Time(HH:MM): ");

gets(time);

printf("\n\tEnter Person Name: ");

gets(person);

printf("\n\tEnter Place: ");

gets(place);

printf("\n\tEnter Duration: ");

gets(duration);

printf("\n\tEnter Short Notes: ");

gets(notes);

for(i = 0; i < strlen(variable\_name\_Time); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Time[i]+508);

}

for(i = 0; i < strlen(time); i++){

fprintf(open\_schedule\_file, "%c", time[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Person); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Person[i]+508);

}

for(i = 0; i < strlen(person); i++){

fprintf(open\_schedule\_file, "%c", person[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Place); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Place[i]+508);

}

for(i = 0; i < strlen(place); i++){

fprintf(open\_schedule\_file, "%c", place[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Duration); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Duration[i]+508);

}

for(i = 0; i < strlen(duration); i++){

fprintf(open\_schedule\_file, "%c", duration[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Notes); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Notes[i]+508);

}

for(i = 0; i < strlen(notes); i++){

fprintf(open\_schedule\_file, "%c", notes[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_New\_Line); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_New\_Line[i]+508);

}

printf("\n\tYour Schedule saved successfully.\n\tPress any key to continue..");

getch();

fclose(open\_schedule\_file);

}

}

else{

printf("\n\n\tNew schedule add canceled...\n\tPress any key to continue...");

getch();

}

}

}

void VIEW\_SCHEDULE(void)

{

FILE \*open\_schedule\_file;

char date[200];

char schedule\_file\_direction[1000];

char ch;

int i;

system("cls");

printf("\n\tFor viewing your schedule, please enter the date(DD-MM-YYYY): ");

gets(date);

strcpy(schedule\_file\_direction, current\_directory);

strcat(schedule\_file\_direction, "\\database\\daily\_schedule\\");

strcat(schedule\_file\_direction, date);

open\_schedule\_file = fopen(schedule\_file\_direction, "r");

if(open\_schedule\_file == NULL){

printf("\n\tIn the database, there is no schedule for this date.\n\tPress any key to continue...");

getch();

}

else{

printf("\n\tSchedule for this date:\n");

while(1){

ch = fgetc(open\_schedule\_file);

if(ch == EOF){

break;

}

else{

printf("%c", ch-508);

}

}

printf("\n\n\tPress any key to continue..");

getch();

fclose(open\_schedule\_file);

}

}

void EDIT\_SCHEDULE(void)

{

FILE \*open\_schedule\_file;

char date[100];

char schedule\_file\_direction[1000];

char ch;

char time[200];

char person[200];

char place[200];

char duration[200];

char notes[1000];

char variable\_name\_Time[] = "\tTime: ";

char variable\_name\_Person[] = "\n\tPerson: ";

char variable\_name\_Place[] = "\n\tPlace: ";

char variable\_name\_Duration[] = "\n\tDuration: ";

char variable\_name\_Notes[] = "\n\tNotes: ";

char variable\_name\_New\_Line[] = "\n\n";

int i;

system("cls");

printf("\n\tFor editing your schedule, please enter the date(DD-MM-YYYY): ");

gets(date);

strcpy(schedule\_file\_direction, current\_directory);

strcat(schedule\_file\_direction, "\\database\\daily\_schedule\\");

strcat(schedule\_file\_direction, date);

open\_schedule\_file = fopen(schedule\_file\_direction, "r");

if(open\_schedule\_file == NULL){

printf("\n\tIn the database, there is no schedule for this date.\n\tPress any key to continue...");

getch();

}

else{

printf("\n\tPrevious schedule for this date:\n");

while(1){

ch = fgetc(open\_schedule\_file);

if(ch == EOF){

break;

}

else{

printf("%c", ch-508);

}

}

fclose(open\_schedule\_file);

printf("\n\n\tNow Enter new schedule:\n");

printf("\n\tEnter Time(HH:MM): ");

gets(time);

printf("\n\tEnter Person Name: ");

gets(person);

printf("\n\tEnter Place: ");

gets(place);

printf("\n\tEnter Duration: ");

gets(duration);

printf("\n\tEnter Short Notes: ");

gets(notes);

printf("\n\tAre you sure to edit this schedule?(Y/N): ");

ch = getch();

if(ch == 'Y' || ch == 'y'){

IDENTITY\_VERIFY();

open\_schedule\_file = fopen(schedule\_file\_direction, "w");

if(open\_schedule\_file == NULL){

printf("\n\tUnable to edit this schedule.\n\tPress any key to continue...");

getch();

}

else{

for(i = 0; i < strlen(variable\_name\_Time); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Time[i]+508);

}

for(i = 0; i < strlen(time); i++){

fprintf(open\_schedule\_file, "%c", time[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Person); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Person[i]+508);

}

for(i = 0; i < strlen(person); i++){

fprintf(open\_schedule\_file, "%c", person[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Place); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Place[i]+508);

}

for(i = 0; i < strlen(place); i++){

fprintf(open\_schedule\_file, "%c", place[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Duration); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Duration[i]+508);

}

for(i = 0; i < strlen(duration); i++){

fprintf(open\_schedule\_file, "%c", duration[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_Notes); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_Notes[i]+508);

}

for(i = 0; i < strlen(notes); i++){

fprintf(open\_schedule\_file, "%c", notes[i] + 508);

}

for(i = 0; i < strlen(variable\_name\_New\_Line); i++){

fprintf(open\_schedule\_file, "%c", variable\_name\_New\_Line[i]+508);

}

printf("\n\tYour Schedule edited successfully.\n\tPress any key to continue..");

getch();

fclose(open\_schedule\_file);

}

}

}

}

void DELETE\_SCHEDULE(void)

{

FILE \*open\_schedule\_file;

char date[200];

char schedule\_file\_direction[1000];

char ch;

int i, status;

system("cls");

printf("\n\tFor viewing your schedule, please enter the date(DD-MM-YYYY): ");

gets(date);

strcpy(schedule\_file\_direction, current\_directory);

strcat(schedule\_file\_direction, "\\database\\daily\_schedule\\");

strcat(schedule\_file\_direction, date);

open\_schedule\_file = fopen(schedule\_file\_direction, "r");

if(open\_schedule\_file == NULL){

printf("\n\tIn the database, there is no schedule for this date.\n\tPress any key to continue...");

getch();

}

else{

printf("\n\tSchedule for this date:\n");

while(1){

ch = fgetc(open\_schedule\_file);

if(ch == EOF){

break;

}

else{

printf("%c", ch-508);

}

}

fclose(open\_schedule\_file);

printf("\n\n\tAre you sure you want to delete this Schedule? (Y/N): ");

ch = getch();

if(ch == 'Y' || ch == 'y'){

IDENTITY\_VERIFY();

status = remove(schedule\_file\_direction);

if(status == 0){

printf("\n\tFile Deleted successfully..\n\tPress any key to continue...");

getch();

}

else{

printf("\n\tFile Delete failed..\n\tPress any key to try again...");

getch();

}

}

else{

printf("\n\tDelete canceled...\n\tPress any key to continue..");

getch();

}

}

}

void ADD\_CLIENT\_INFORMATION(void)

{

FILE \*open\_client\_info\_file;

char name[100];

char profession[100];

char number[100];

char email[200];

char address\_office[200];

char address\_home[200];

char notes[1000];

char client\_information\_file\_direction[1000];

int i;

system("cls");

printf("\n\tPlease enter your client Name: ");

gets(name);

strcpy(client\_information\_file\_direction, current\_directory);

strcat(client\_information\_file\_direction, "\\database\\client\_information\\");

strcat(client\_information\_file\_direction, name);

open\_client\_info\_file = fopen(client\_information\_file\_direction, "r");

if(open\_client\_info\_file == NULL){

open\_client\_info\_file = fopen(client\_information\_file\_direction, "w");

if(open\_client\_info\_file == NULL){

printf("\n\tUnable to create file...\n\tPlease try again...");

getch();

}

else{

printf("\n\tProfession: ");

gets(profession);

printf("\n\tPhone Number: ");

gets(number);

printf("\n\tE-Mail: ");

gets(email);

printf("\n\tOffice Address: ");

gets(address\_office);

printf("\n\tHome Address: ");

gets(address\_home);

printf("\n\tShort Notes: ");

gets(notes);

for(i = 0; i < strlen(profession); i++){

fprintf(open\_client\_info\_file, "%c", profession[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(number); i++){

fprintf(open\_client\_info\_file, "%c", number[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(email); i++){

fprintf(open\_client\_info\_file, "%c", email[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(address\_office); i++){

fprintf(open\_client\_info\_file, "%c", address\_office[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(address\_home); i++){

fprintf(open\_client\_info\_file, "%c", address\_home[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(notes); i++){

fprintf(open\_client\_info\_file, "%c", notes[i] + 508);

}

fclose(open\_client\_info\_file);

printf("\n\tClient Information saved successfully.\n\tPress any key to continue....");

getch();

}

}

else{

printf("\n\tIt seems that you already have client information by this name.\n\tYou can't add two client information by the same name.\n\tPlease add a different name.\n\tPress any key to continue...");

getch();

fclose(open\_client\_info\_file);

}

}

void VIEW\_CLIENT\_INFORMATION(void)

{

FILE \*open\_client\_info\_file;

char name[100];

char profession[200];

char number[200];

char email[200];

char address\_office[200];

char address\_home[200];

char notes[1000];

char client\_information\_file\_direction[1000];

int i;

system("cls");

printf("\n\tFor viewing information, please enter your client name: ");

gets(name);

strcpy(client\_information\_file\_direction, current\_directory);

strcat(client\_information\_file\_direction, "\\database\\client\_information\\");

strcat(client\_information\_file\_direction, name);

open\_client\_info\_file = fopen(client\_information\_file\_direction, "r");

if(open\_client\_info\_file == NULL){

printf("\n\tIn the database, there is no client information by this name..\n\tPress any key to try again...");

getch();

}

else{

fgets(profession, 200, open\_client\_info\_file);

for(i = 0; i < strlen(profession); i++){

profession[i] = profession[i] - 508;

}

profession[i-1] = '\0';

fgets(number, 200, open\_client\_info\_file);

for(i = 0; i < strlen(number); i++){

number[i] = number[i] - 508;

}

number[i-1] = '\0';

fgets(email, 200, open\_client\_info\_file);

for(i = 0; i < strlen(email); i++){

email[i] = email[i] - 508;

}

email[i-1] = '\0';

fgets(address\_office, 200, open\_client\_info\_file);

for(i = 0; i < strlen(address\_office); i++){

address\_office[i] = address\_office[i] - 508;

}

address\_office[i-1] = '\0';

fgets(address\_home, 200, open\_client\_info\_file);

for(i = 0; i < strlen(address\_home); i++){

address\_home[i] = address\_home[i] - 508;

}

address\_home[i-1] = '\0';

fgets(notes, 200, open\_client\_info\_file);

for(i = 0; i < strlen(notes); i++){

notes[i] = notes[i] - 508;

}

printf("\n\tProfession: %s\n\tPhone Number: %s\n\tE-Mail: %s\n\tOffice Address: %s\n\tHome Address: %s\n\tShort Notes: %s\n\n\tPress any key to continue..", profession, number, email, address\_office, address\_home, notes);

getch();

fclose(open\_client\_info\_file);

}

}

void EDIT\_CLIENT\_INFORMATION(void)

{

FILE \*open\_client\_info\_file;

char name[100];

char profession[200];

char number[200];

char email[200];

char address\_office[200];

char address\_home[200];

char notes[1000];

char ch;

char client\_information\_file\_direction[1000];

int i;

system("cls");

printf("\n\tFor viewing information, please enter your client name: ");

gets(name);

strcpy(client\_information\_file\_direction, current\_directory);

strcat(client\_information\_file\_direction, "\\database\\client\_information\\");

strcat(client\_information\_file\_direction, name);

open\_client\_info\_file = fopen(client\_information\_file\_direction, "r");

if(open\_client\_info\_file == NULL){

printf("\n\tIn the database, there is no client information by this name..\n\tPress any key to try again...");

getch();

}

else{

fgets(profession, 200, open\_client\_info\_file);

for(i = 0; i < strlen(profession); i++){

profession[i] = profession[i] - 508;

}

profession[i-1] = '\0';

fgets(number, 200, open\_client\_info\_file);

for(i = 0; i < strlen(number); i++){

number[i] = number[i] - 508;

}

number[i-1] = '\0';

fgets(email, 200, open\_client\_info\_file);

for(i = 0; i < strlen(email); i++){

email[i] = email[i] - 508;

}

email[i-1] = '\0';

fgets(address\_office, 200, open\_client\_info\_file);

for(i = 0; i < strlen(address\_office); i++){

address\_office[i] = address\_office[i] - 508;

}

address\_office[i-1] = '\0';

fgets(address\_home, 200, open\_client\_info\_file);

for(i = 0; i < strlen(address\_home); i++){

address\_home[i] = address\_home[i] - 508;

}

address\_home[i-1] = '\0';

fgets(notes, 200, open\_client\_info\_file);

for(i = 0; i < strlen(notes); i++){

notes[i] = notes[i] - 508;

}

fclose(open\_client\_info\_file);

printf("\n\tProfession: %s\n\tPhone Number: %s\n\tE-Mail: %s\n\tOffice Address: %s\n\tHome Address: %s\n\tShort Notes: %s\n\n\tWhich information do you want to edit?\n\t=> 1. Profession\n\t=> 2. Number\n\t=> 3. E-Mail\n\t=> 4. Office Address\n\t=> 5. Home Address\n\t=> 6. Short Notes\n\t=> 7. Whole Information\n\t=> 8. Cancel\n\n\tEnter Command: ", profession, number, email, address\_office, address\_home, notes);

ch = getch();

if(ch == '1'){

printf("\n\tEnter Profession: ");

gets(profession);

}

else if(ch == '2'){

printf("\n\tEnter Phone Number: ");

gets(number);

}

else if(ch == '3'){

printf("\n\tEnter E-Mail: ");

gets(email);

}

else if(ch == '4'){

printf("\n\tEnter Office Address: ");

gets(address\_office);

}

else if(ch == '5'){

printf("\n\tEnter Home Address: ");

gets(address\_home);

}

else if(ch == '6'){

printf("\n\tEnter Short Notes: ");

gets(notes);

}

else if(ch == '7'){

printf("\n\tProfession: ");

gets(profession);

printf("\n\tPhone Number: ");

gets(number);

printf("\n\tE-Mail: ");

gets(email);

printf("\n\tOffice Address: ");

gets(address\_office);

printf("\n\tHome Address: ");

gets(address\_home);

printf("\n\tShort Notes: ");

gets(notes);

}

else{

printf("\n\tEdit canceled....\n\tPress any key to continue...");

getch();

}

if(ch >= '1' && ch <= '7'){

printf("\n\tAre you sure to edit this information?(Y/N): ");

ch = getch();

if(ch == 'Y' || ch == 'y'){

IDENTITY\_VERIFY();

open\_client\_info\_file = fopen(client\_information\_file\_direction, "w");

if(open\_client\_info\_file == NULL){

printf("\n\tUnable to edit information.... Please try again...");

getch();

}

else{

for(i = 0; i < strlen(profession); i++){

fprintf(open\_client\_info\_file, "%c", profession[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(number); i++){

fprintf(open\_client\_info\_file, "%c", number[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(email); i++){

fprintf(open\_client\_info\_file, "%c", email[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(address\_office); i++){

fprintf(open\_client\_info\_file, "%c", address\_office[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(address\_home); i++){

fprintf(open\_client\_info\_file, "%c", address\_home[i] + 508);

}

fprintf(open\_client\_info\_file, "\n");

for(i = 0; i < strlen(notes); i++){

fprintf(open\_client\_info\_file, "%c", notes[i] + 508);

}

fclose(open\_client\_info\_file);

printf("\n\tEdited Successfully..\n\tPress any key to continue...");

getch();

}

}

else{

printf("\n\tEdit canceled....\n\tPress any key to continue...");

getch();

}

}

}

}

void DELETE\_CLIENT\_INFORMATION(void)

{

FILE \*open\_client\_info\_file;

char name[100];

char profession[200];

char number[200];

char email[200];

char address\_office[200];

char address\_home[200];

char notes[1000];

char ch;

char client\_information\_file\_direction[1000];

int i, status;

system("cls");

printf("\n\tFor deleting information, please enter your client name: ");

gets(name);

strcpy(client\_information\_file\_direction, current\_directory);

strcat(client\_information\_file\_direction, "\\database\\client\_information\\");

strcat(client\_information\_file\_direction, name);

open\_client\_info\_file = fopen(client\_information\_file\_direction, "r");

if(open\_client\_info\_file == NULL){

printf("\n\tIn the database, there is no client information by this name..\n\tPress any key to try again...");

getch();

}

else{

fgets(profession, 200, open\_client\_info\_file);

for(i = 0; i < strlen(profession); i++){

profession[i] = profession[i] - 508;

}

profession[i-1] = '\0';

fgets(number, 200, open\_client\_info\_file);

for(i = 0; i < strlen(number); i++){

number[i] = number[i] - 508;

}

number[i-1] = '\0';

fgets(email, 200, open\_client\_info\_file);

for(i = 0; i < strlen(email); i++){

email[i] = email[i] - 508;

}

email[i-1] = '\0';

fgets(address\_office, 200, open\_client\_info\_file);

for(i = 0; i < strlen(address\_office); i++){

address\_office[i] = address\_office[i] - 508;

}

address\_office[i-1] = '\0';

fgets(address\_home, 200, open\_client\_info\_file);

for(i = 0; i < strlen(address\_home); i++){

address\_home[i] = address\_home[i] - 508;

}

address\_home[i-1] = '\0';

fgets(notes, 200, open\_client\_info\_file);

for(i = 0; i < strlen(notes); i++){

notes[i] = notes[i] - 508;

}

fclose(open\_client\_info\_file);

printf("\n\tProfession: %s\n\tPhone Number: %s\n\tE-Mail: %s\n\tOffice Address: %s\n\tHome Address: %s\n\tShort Notes: %s\n\n\tAre you sure to delete this client information?(Y/N): ", profession, number, email, address\_office, address\_home, notes);

ch = getch();

if(ch == 'Y' || ch == 'y'){

IDENTITY\_VERIFY();

status = remove(client\_information\_file\_direction);

if(status == 0){

printf("\n\tClient information deleted successfully..\n\tPress any key to continue..");

getch();

}

else{

printf("\n\tClient information deletion failed...\n\tPress any key to continue...");

getch();

}

}

else{

printf("\n\tDelete canceled...\n\tPress any key to continue...");

getch();

}

}

}

void ABOUT(void)

{

system("cls");

WELCOME\_SCREEN();

printf("\n\n\tVersion: 01.07\n\tLast Updated: 24.04.2022\n\n\tPowered by SfM1");

printf("\n\n\tTeam: TUF CODER\n\n\tTeam Members:");

printf("\n\tJoy Kumar Ghosh\n\tID: 2211424 6 42");

printf("\n\n\tKazi Abid Shahoriar\n\tID: 2211967 0 42");

printf("\n\n\tJannatul Mawa Tahi\n\tID: 2212096 0 42");

printf("\n\n\tFarhana Rahman Risha\n\tID: 2211598 6 42");

printf("\n\n\tKhandaker Anjuman Parvez\n\tID: 2212536 0 42");

getch();

EXIT\_SCREEN();

}

void WELCOME\_SCREEN(void)

{

printf("\t \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

printf("\n\t.-/| \\ / |\\-.");

printf("\n\t|||| | ||||");

printf("\n\t|||| Welcome to | ||||");

printf("\n\t|||| | ||||");

printf("\n\t||||Password Protected | ||||");

printf("\n\t|||| | V. 01.04 ||||");

printf("\n\t|||| Personal Diary | Powered by SfM1 ||||");

printf("\n\t|||| | ||||");

printf("\n\t|||| with Encryption | ||||");

printf("\n\t|||| | ||||");

printf("\n\t|||| | ||||");

printf("\n\t||||\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_||||");

printf("\n\t||/===================\\|/===================\\||");

printf("\n\t`--------------------~\_\_\_~-------------------''");

}

void HEADER\_SCREEN(char menu\_msg[])

{

int lenght\_menu\_msg = strlen(menu\_msg), i, j;

printf("\n\n\t");

for(i = 0; i <= (45-lenght\_menu\_msg-2)/2; i++){

printf("\xB2");

}

printf(" %s ", menu\_msg);

for(j = 0; j <= (44-lenght\_menu\_msg-i); j++){

printf("\xB2");

}

printf("\n");

}

void EXIT\_SCREEN(void)

{

system("cls");

printf("\n\t \_ .-\"--.\_");

printf("\n\t / \\ / \_\_\_\_\\");

printf("\n\t ||\\\\ / /`(");

printf("\n\t || \\\\ \_| '``'-.");

printf("\n\t | \\\_\\\\ ` 9\\ ,");

printf("\n\t \\\_ 9 \_ '-.= .--'|}");

printf("\n\t | \_ \\) | / /}}");

printf("\n\t \\/ = \\ ;\_.'/ .=\\.--'`\\}");

printf("\n\t | `-`\_\_.;---.//` '---./'");

printf("\n\t '.\_\_\_..-'` `| \_\_\_ \_ \_ \_\_\_ \_ \_ \_ \_");

printf("\n\t \_/ \_\_.-.\_\_/ |\_ \_| | | | / \_ \\ | \\| | | |/ /");

printf("\n\t .-' .-' ||| | | | | | | | ` | | <");

printf("\n\t .-/ / |\\\\ |\_| |\_|\_| |\_|\_| |\_|\\\_| |\_|\\\_\\");

printf("\n\t{ | /\_ / \\| \_ \_ \_\_\_ \_ \_");

printf("\n\t `-\\ `\\--;` | | | / \\ | | |");

printf("\n\t '-. | | \\ / | | | | | |");

printf("\n\t ) / \_/ |\_| \\\_\_\_/ \\\_\_\_/");

printf("\n\t / \_\_.' '--.");

printf("\n\t ( '--. \_\_\_)))");

printf("\n\t `-..\_\_\_\_)))");

printf("\n\n\t\t\t\t\tFOR USING OUR SOFTWARE\n\n\n");

getch();

}

# Team Info.:

Team Name: TUF CODER

Team Members:

Joy Kumar Ghosh ID: 2211424 6 42

Kazi Abid Shahoriar ID: 2211967 0 42

Jannatul Mawa Tahi ID: 2212096 0 42

Farhana Rahman Risha ID: 2211598 6 42

Khandaker Anjuman Parvez ID: 2212536 0 42

**Thank You**