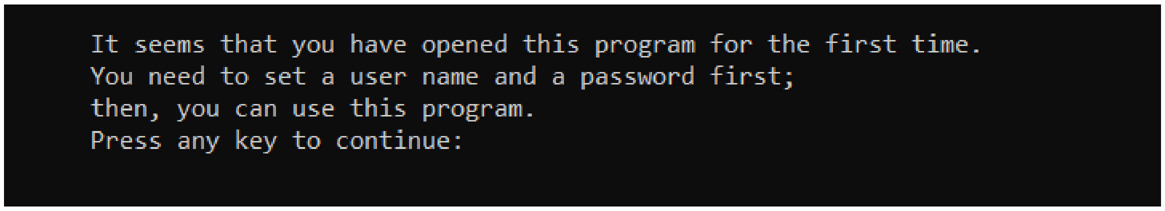
Personal Diary Management System

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

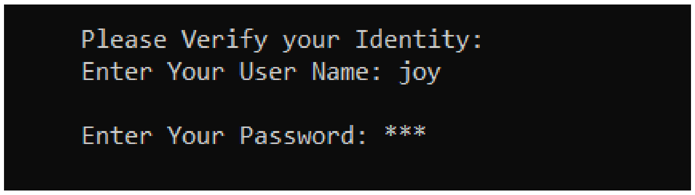
How to run:

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



```
It seems that you have opened this program for the first time.
You need to set a user name and a password first;
then, you can use this program.
Press any key to continue:
```

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



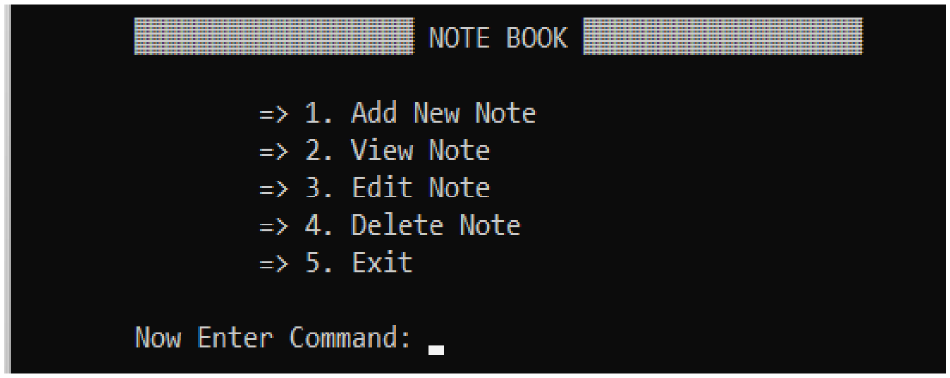
```
Please Verify your Identity:
Enter Your User Name: joy

Enter Your Password: ***
```

In verification, password will not be shown.

Key Features

1. Notebooks

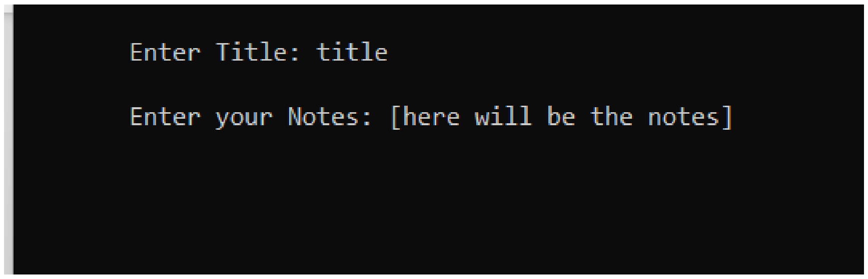
This function is for managing your personal notes. Here we have four options. The first one is for adding new notes, second one is for viewing added notes, third one is for editing existing notes, and the last one for deleting previously added notes.



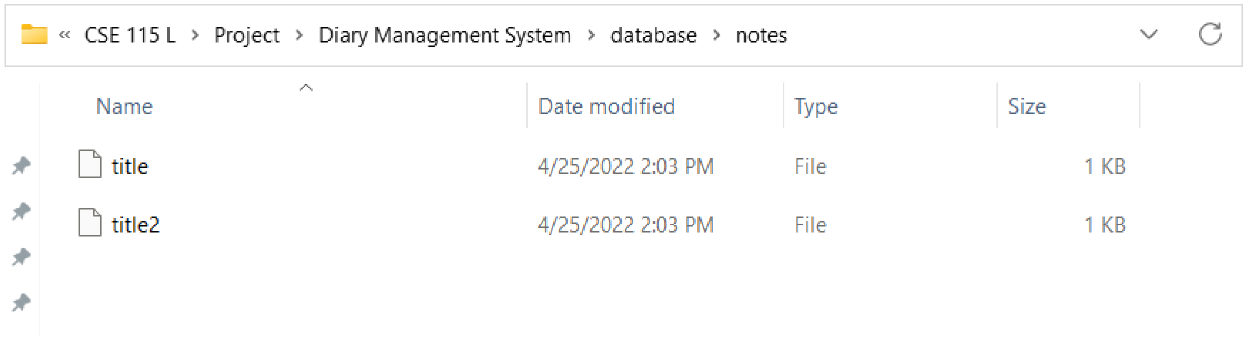
```
░░░░░░░░░░░░░░░░ NOTE BOOK ░░░░░░░░░░░░░░░░

        => 1. Add New Note
        => 2. View Note
        => 3. Edit Note
        => 4. Delete Note
        => 5. Exit

Now Enter Command: _
```

a. 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.



```
Enter Title: title

Enter your Notes: [here will be the notes]
```

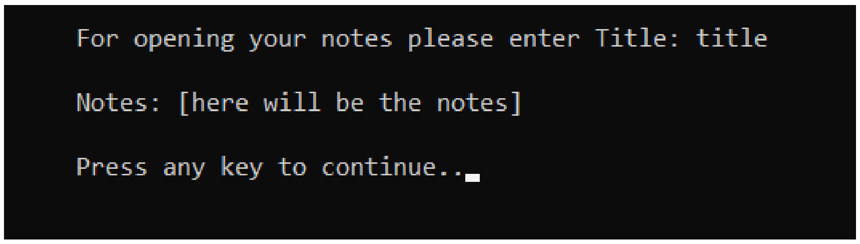


« CSE 115 L › Project › Diary Management System › database › notes

Name	Date modified	Type	Size
title	4/25/2022 2:03 PM	File	1 KB
title2	4/25/2022 2:03 PM	File	1 KB

b. View Notes

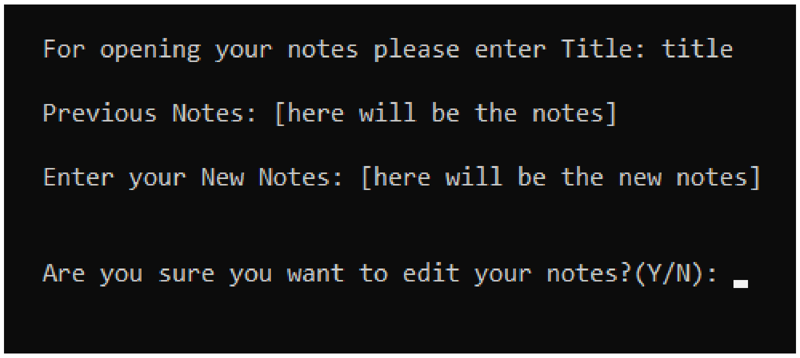
Here, user just needs to enter the title of the notes which was previously added. If the given title is available in database, then the console will show the notes by decrypting stored data.



```
For opening your notes please enter Title: title

Notes: [here will be the notes]

Press any key to continue..
```

c. Edit Notes

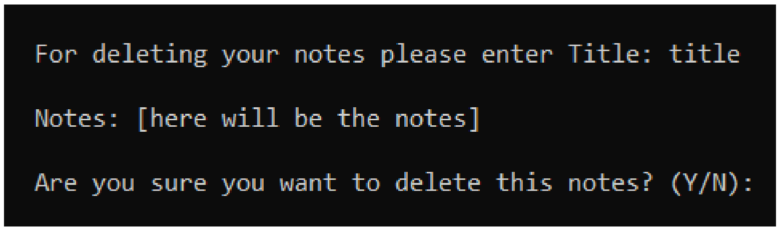
It's an interesting feature that can edit user's previously added notes. First, it will display the existing notes that, user want to edit. Then, it will ask for new data that, user want to change. After that console will ask for a confirmation and an identity verification will be displayed.



```
For opening your notes please enter Title: title

Previous Notes: [here will be the notes]

Enter your New Notes: [here will be the new notes]


Are you sure you want to edit your notes?(Y/N):
```

d. Delete Notes

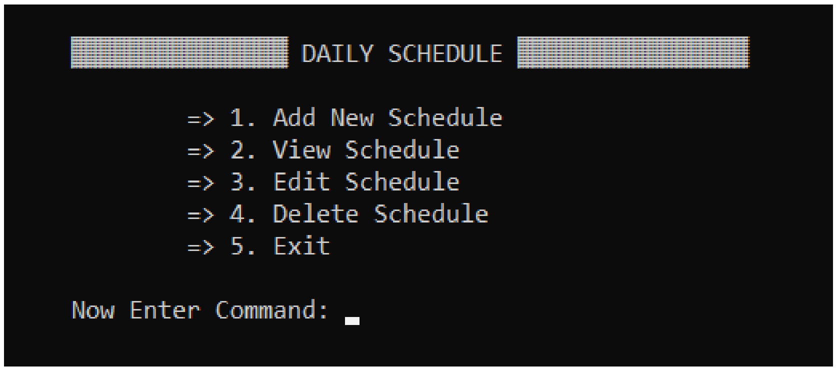
This option is specifically used for deleting previously added notes. User will be asked for the title only. Console will show the notes and ask for a confirmation.



```
For deleting your notes please enter Title: title

Notes: [here will be the notes]

Are you sure you want to delete this notes? (Y/N):
```

2. Daily Schedule

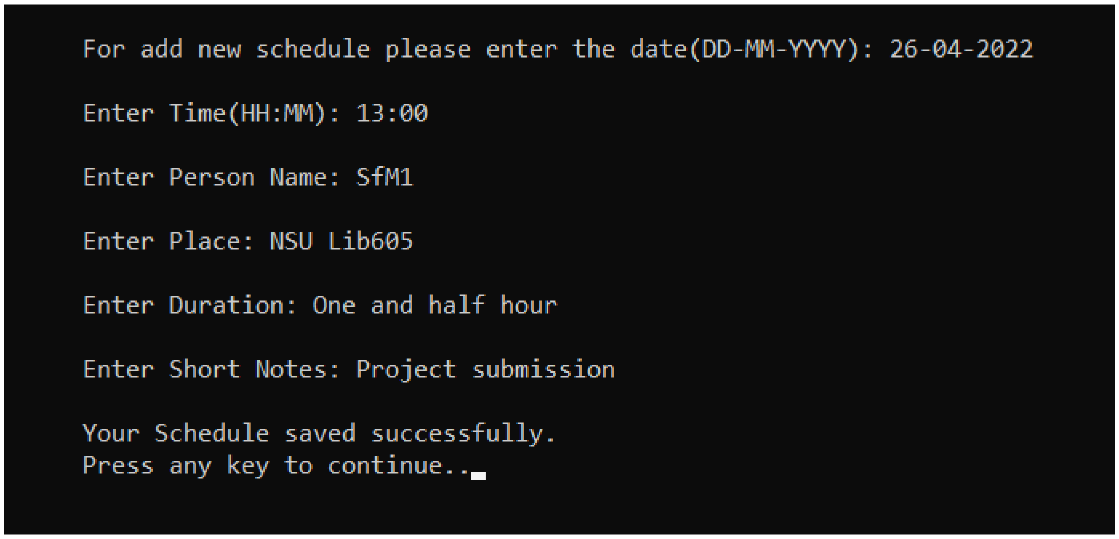
Daily schedule is used for user’s daily time management necessities. Its features are given below.



```
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ DAILY SCHEDULE ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒

        => 1. Add New Schedule
        => 2. View Schedule
        => 3. Edit Schedule
        => 4. Delete Schedule
        => 5. Exit

Now Enter Command: _
```

a. Add New Schedule

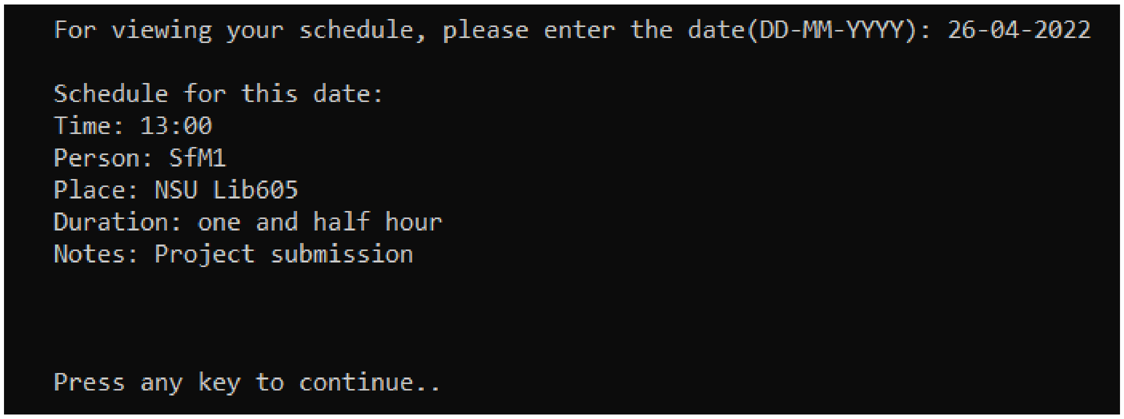
In this feature, user can easily add daily time schedule for meetings, works etc. User will be asked to enter date, time, person name, place, duration and short notes. Schedule will be stored in a file named by the dates with encryption.



```
For add new schedule please enter the date(DD-MM-YYYY): 26-04-2022

Enter Time(HH:MM): 13:00

Enter Person Name: SfM1

Enter Place: NSU Lib605

Enter Duration: One and half hour

Enter Short Notes: Project submission

Your Schedule saved successfully.
Press any key to continue.._
```

b. View Schedule

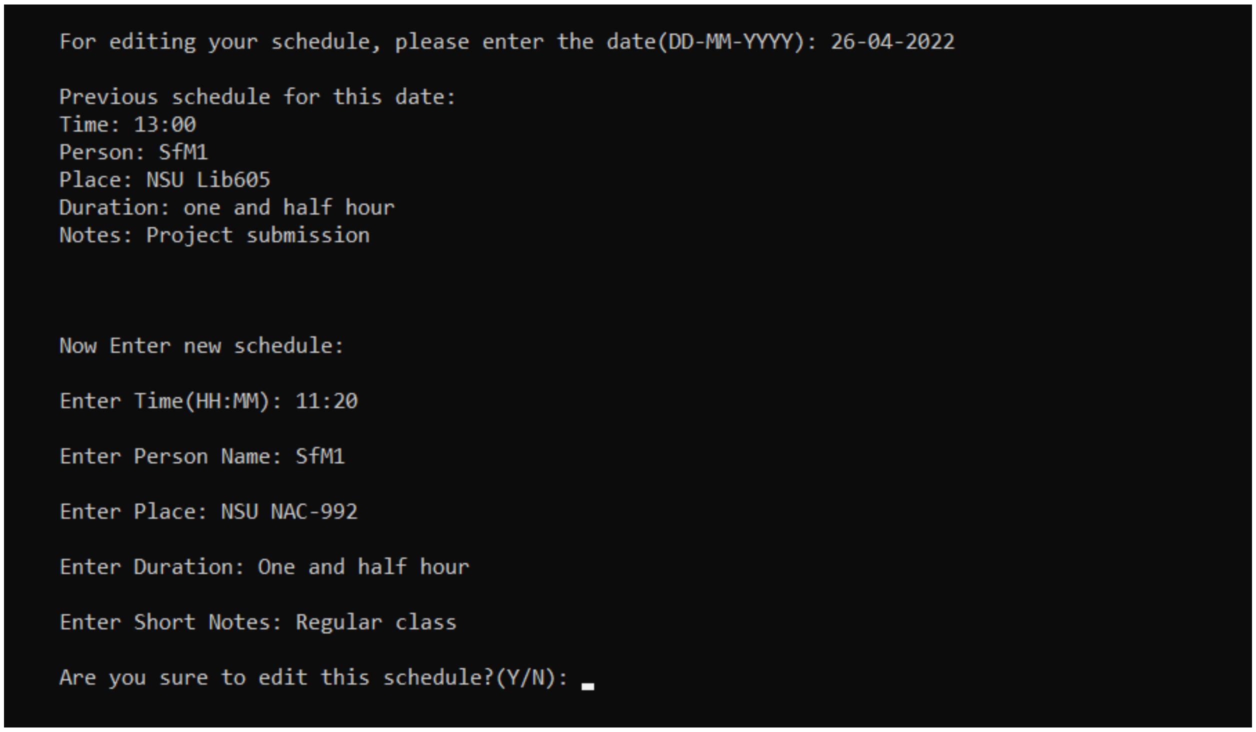
This feature will display all the schedule according to dates. It will ask user to enter the date only.



```
For viewing your schedule, please enter the date(DD-MM-YYYY): 26-04-2022

Schedule for this date:
Time: 13:00
Person: SfM1
Place: NSU Lib605
Duration: one and half hour
Notes: Project submission



Press any key to continue..
```

c. Edit Schedule

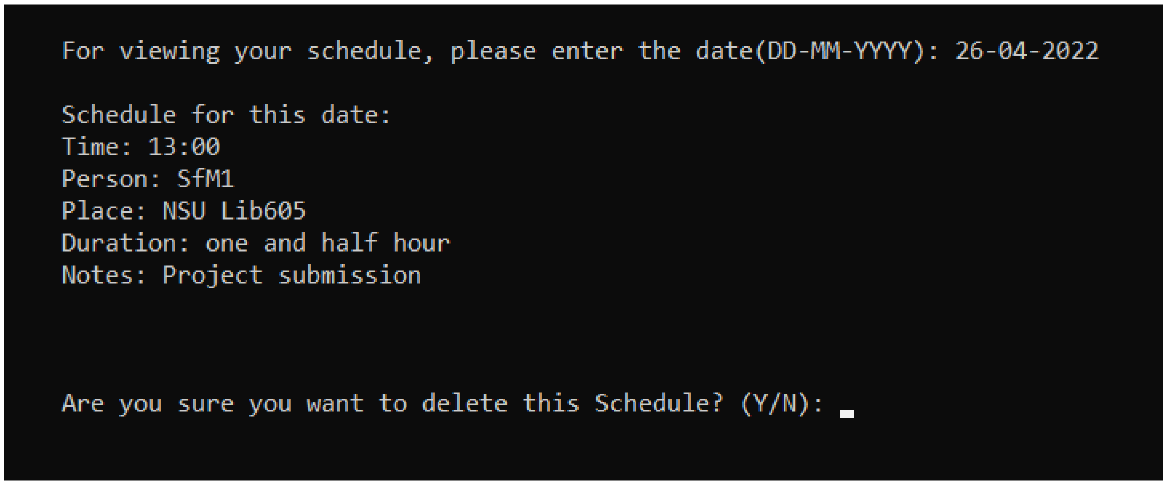
Edit Schedule is used for editing existing time schedules which are added by the user previously. User need to enter whole data again. Then, console will ask for the confirmation.



```
For editing your schedule, please enter the date(DD-MM-YYYY): 26-04-2022

Previous schedule for this date:
Time: 13:00
Person: SfM1
Place: NSU Lib605
Duration: one and half hour
Notes: Project submission



Now Enter new schedule:

Enter Time(HH:MM): 11:20

Enter Person Name: SfM1

Enter Place: NSU NAC-992

Enter Duration: One and half hour

Enter Short Notes: Regular class

Are you sure to edit this schedule?(Y/N):
```

d. Delete Schedule

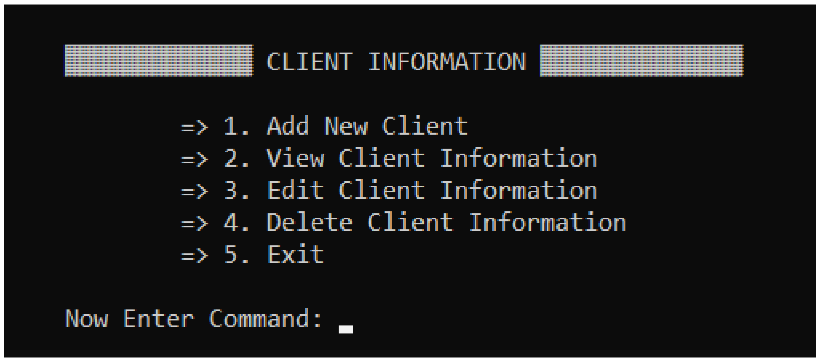
This option is used for deleting user’s previous schedule by date. Console will ask for confirmation and identity verification.



```
For viewing your schedule, please enter the date(DD-MM-YYYY): 26-04-2022

Schedule for this date:
Time: 13:00
Person: SfM1
Place: NSU Lib605
Duration: one and half hour
Notes: Project submission



Are you sure you want to delete this Schedule? (Y/N): _
```

3. Client Information

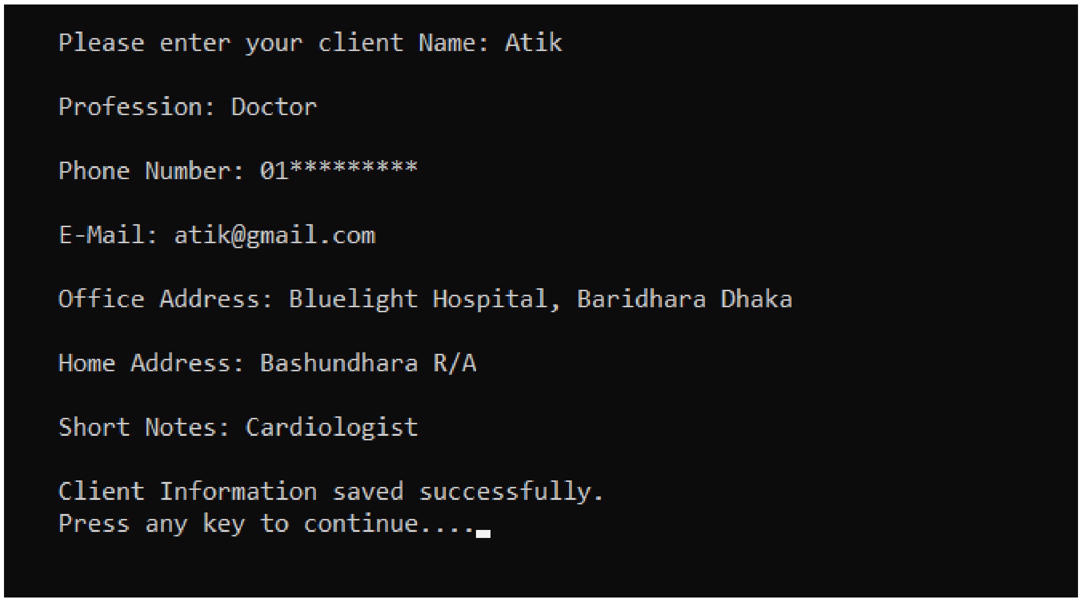
This is the most detailed feature in this whole software. It contains the features given below.



```
▒▒▒▒▒▒▒▒▒▒▒▒▒ CLIENT INFORMATION ▒▒▒▒▒▒▒▒▒▒▒▒▒▒

        => 1. Add New Client
        => 2. View Client Information
        => 3. Edit Client Information
        => 4. Delete Client Information
        => 5. Exit

Now Enter Command: _
```

a. Add New Client

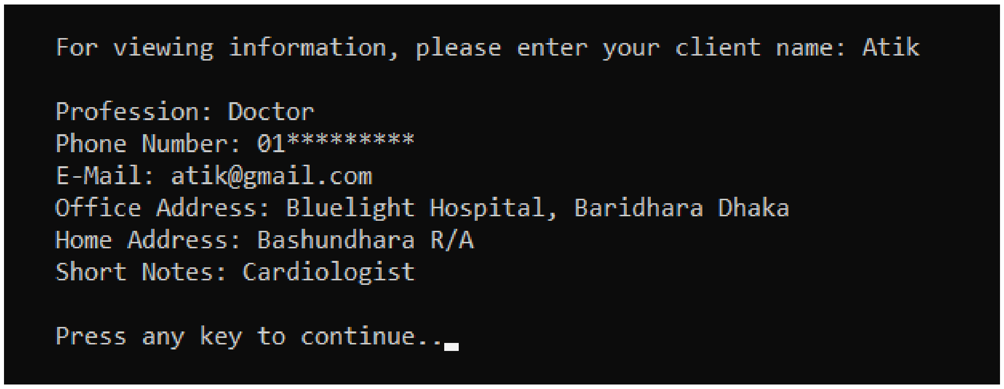
This feature gives user ability to add new clients by their names. Also, adds details such as, profession, phone number, E-mail, office address, home address and short notes. All the information will be sorted by their names with encryption.



```
Please enter your client Name: Atik

Profession: Doctor

Phone Number: 01*********

E-Mail: atik@gmail.com

Office Address: Bluelight Hospital, Baridhara Dhaka

Home Address: Bashundhara R/A

Short Notes: Cardiologist

Client Information saved successfully.
Press any key to continue...._
```

b. View Client Information

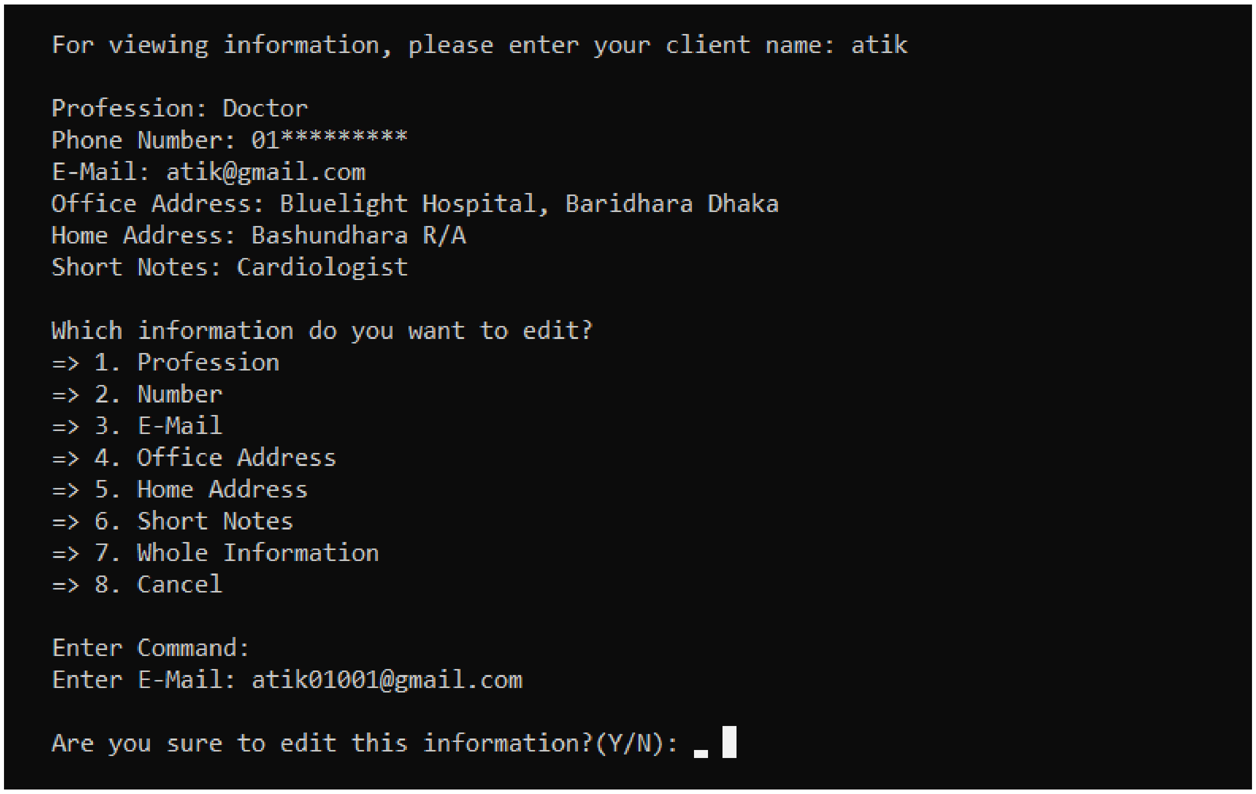
In this feature console will display client details according to client names.



```
For viewing information, please enter your client name: Atik

Profession: Doctor
Phone Number: 01*********
E-Mail: atik@gmail.com
Office Address: Bluelight Hospital, Baridhara Dhaka
Home Address: Bashundhara R/A
Short Notes: Cardiologist

Press any key to continue..
```

c. Edit Client Information

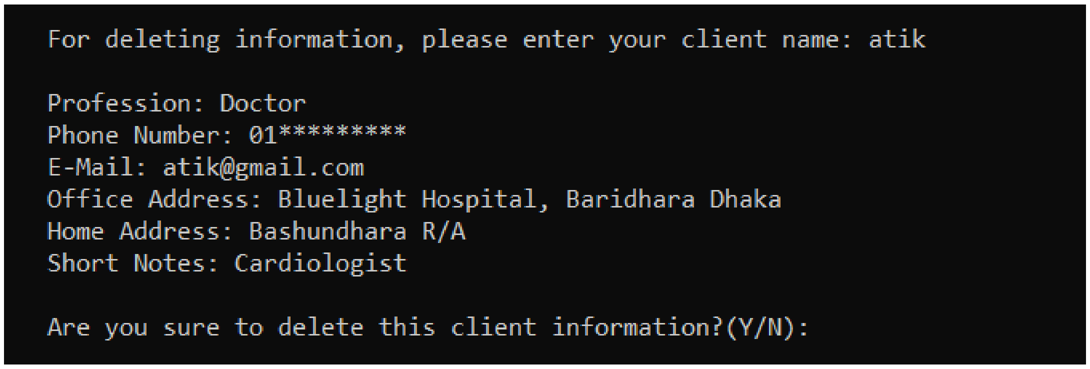
If user want to edit any client information, this option will help to do that. User can edit a specific field or the whole information by some simple commands.
At the end, console will ask for confirmation and identity verification.



```
For viewing information, please enter your client name: atik

Profession: Doctor
Phone Number: 01*********
E-Mail: atik@gmail.com
Office Address: Bluelight Hospital, Baridhara Dhaka
Home Address: Bashundhara R/A
Short Notes: Cardiologist

Which information do you want to edit?
=> 1. Profession
=> 2. Number
=> 3. E-Mail
=> 4. Office Address
=> 5. Home Address
=> 6. Short Notes
=> 7. Whole Information
=> 8. Cancel

Enter Command:
Enter E-Mail: atik01001@gmail.com

Are you sure to edit this information?(Y/N):
```

d. Delete Client Information

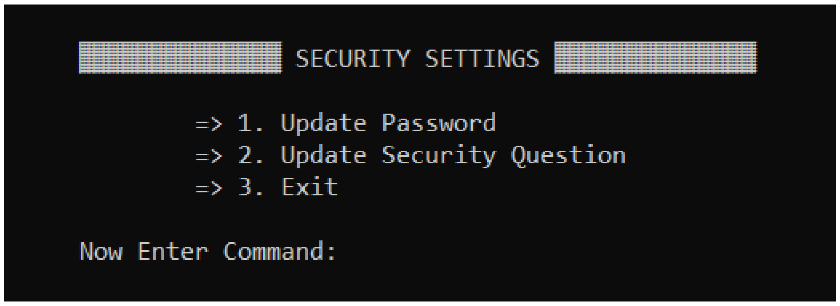
This option is useful for deleting existing client information. Console will ask user for confirmation and verification after entering the client’s name.



```
For deleting information, please enter your client name: atik

Profession: Doctor
Phone Number: 01*********
E-Mail: atik@gmail.com
Office Address: Bluelight Hospital, Baridhara Dhaka
Home Address: Bashundhara R/A
Short Notes: Cardiologist

Are you sure to delete this client information?(Y/N):
```

4. Security Settings

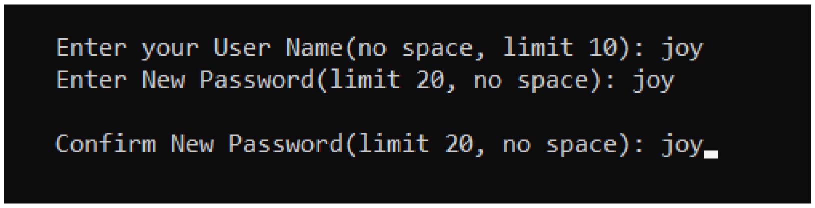
Here, user can manage security settings such as, change password and security question.



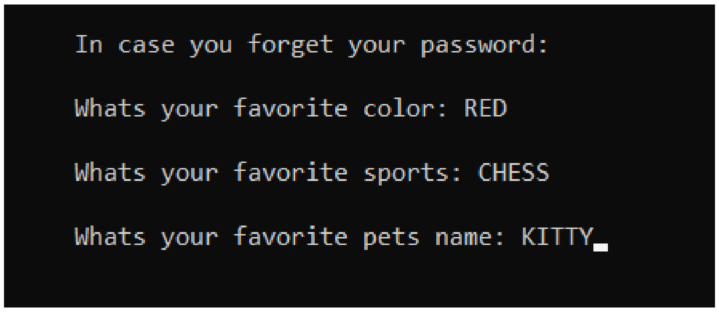
```
  ▒▒▒▒▒▒▒▒▒▒▒▒▒▒ SECURITY SETTINGS ▒▒▒▒▒▒▒▒▒▒▒▒▒▒

          => 1. Update Password
          => 2. Update Security Question
          => 3. Exit

  Now Enter Command:
```

a. 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.



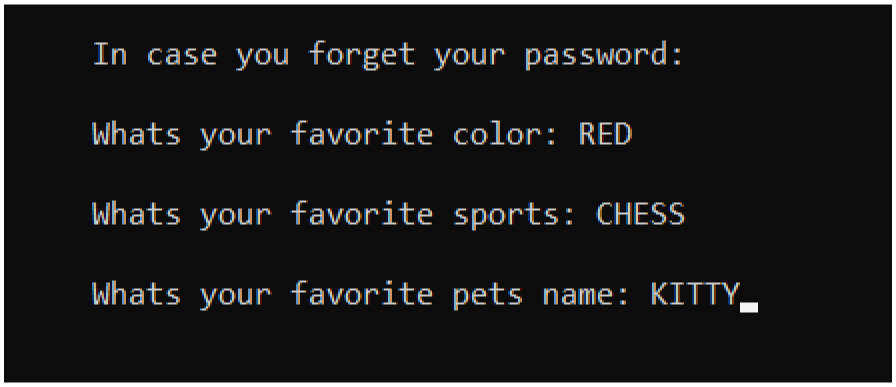
```
Enter your User Name(no space, limit 10): joy
Enter New Password(limit 20, no space): joy

Confirm New Password(limit 20, no space): joy_
```



```
In case you forget your password:

Whats your favorite color: RED

Whats your favorite sports: CHESS

Whats your favorite pets name: KITTY_
```

b. Update Security Question

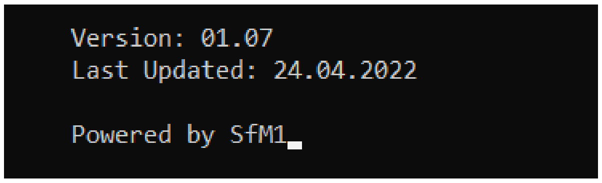
Here, console will ask user to verify identity. Then it will ask for new security question.



```
In case you forget your password:

Whats your favorite color: RED

Whats your favorite sports: CHESS

Whats your favorite pets name: KITTY_
```

5. About Software

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

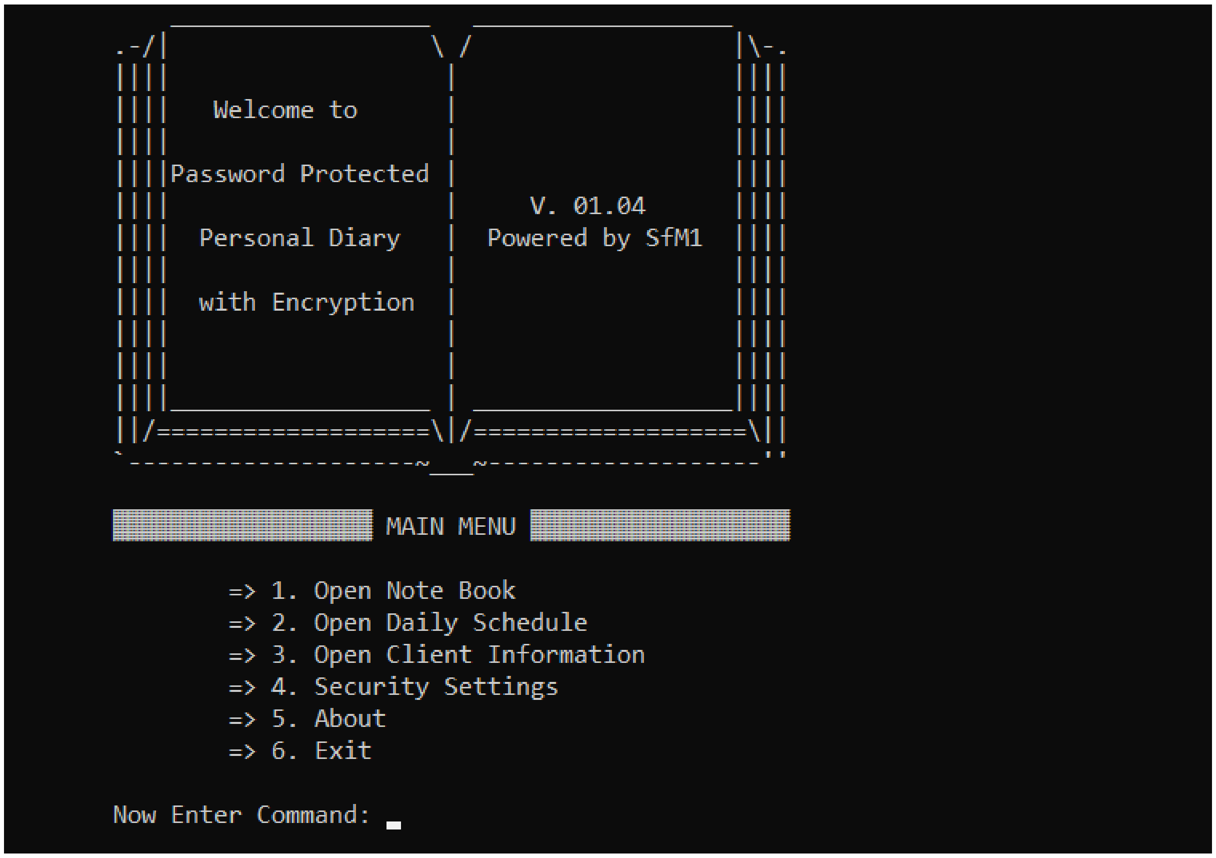


```
Version: 01.07
Last Updated: 24.04.2022

Powered by SfM1_
```

6. UI Design

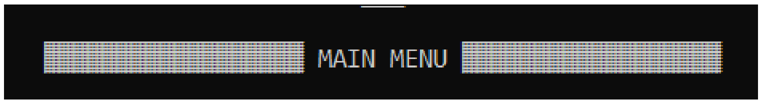
We added some interesting asci arts which are given below.

a. Welcome Screen



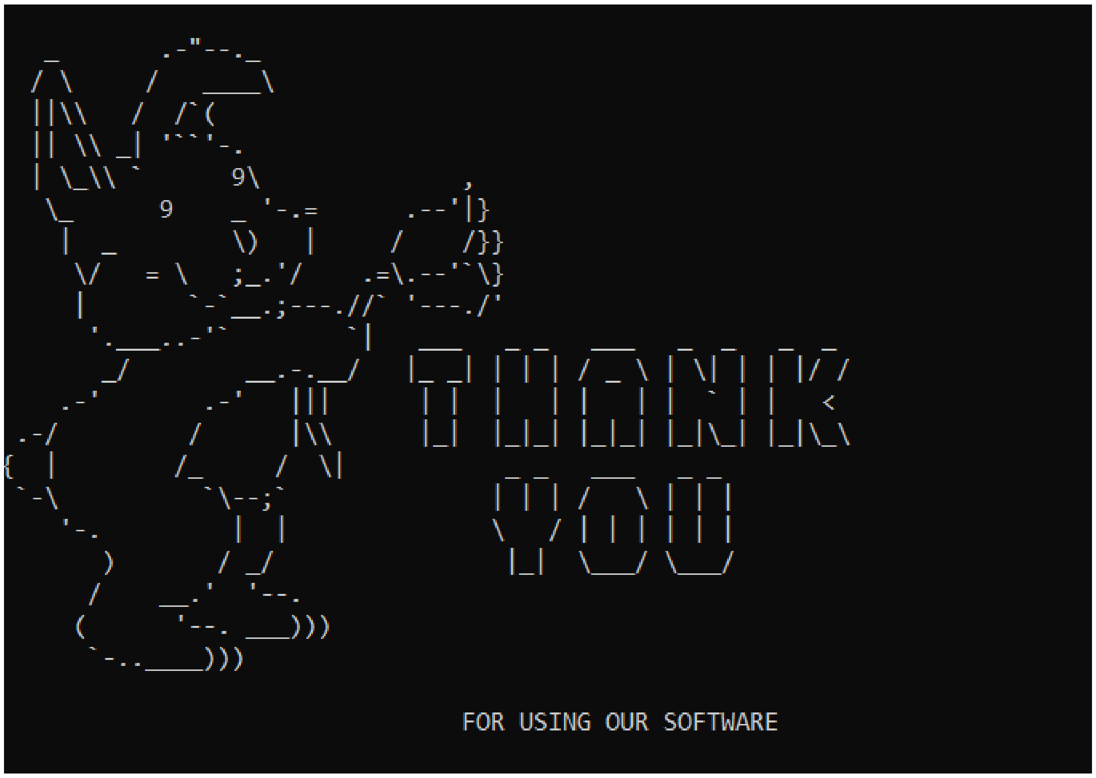
```
    __________________   __________________
.-/|                  \ /                  |\-.
||||                   |                   ||||
||||   Welcome to      |                   ||||
||||                   |                   ||||
||||Password Protected |                   ||||
||||                   |     V. 01.04      ||||
||||  Personal Diary   |  Powered by SfM1  ||||
||||                   |                   ||||
||||  with Encryption  |                   ||||
||||                   |                   ||||
||||                   |                   ||||
||||__________________ | __________________||||
||/===================\|/===================\||
`--------------------~___~-------------------''

▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ MAIN MENU ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒

        => 1. Open Note Book
        => 2. Open Daily Schedule
        => 3. Open Client Information
        => 4. Security Settings
        => 5. About
        => 6. Exit

Now Enter Command: _
```

b. Header Screen



```
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒ MAIN MENU ▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
```

c. Exit Screen



```
THANK
YOU

FOR USING OUR SOFTWARE
```

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