```
#include<iostream.h>
#include<ctype.h>
#include<conio.h>
#include<string.h>
#include<fstream.h>
#include<time.h>
#include<math.h>
#include<stdio.h>
#include<process.h>
#include<dos.h>
const char a=176;
const char b=219;
const char ll=179;
void tp_lgo()
{
        clrscr();
        cout<<"\n\t\t\t"<<a<<a<<a<<a<<a<<"  "<<a<<"  "<<a<<"  "<<a<<a<<a<<a<<a;
        cout<<"  "<<a<<a<<"  "<<a<<"  "<<a<<"  "<<a<<"   ";
        cout<<"\n\t\t\t"<<"  "<<a<<"  "<<"  "<<a<<a<<a<<a<<"  ";
        cout<<"  "<<a<<"  "<<"  "<<a<<" "<<a<<" "<<a<<"  "<<a<<a<<a;
        cout<<"   "<<" Smart ";
        cout<<"\n\t\t\t"<<"  "<<a<<"  "<<"  "<<a<<"  "<<a<<"  "<<a<<a<<a<<a<<a;
        cout<<"  "<<a<<"  "<<a<<a<<"  "<<a<<"  "<<a<<"   ";

}
void std_err1()
{
        cout<<"\n Error : Filed Cannot Be Left Blank -> ";
}
void std_err2()
{
        cout<<"\n Error : Invalid Parameter -> ";
}
void std_err3()
{
        cout<<"\n Press ~ [SHIFT+`] to retain the old one \n";
}
int digit_sqncr(int key_var_digit)
{
        int sum=0;
        int digit=0;
        while(key_var_digit>0)
        {
                digit=key_var_digit%10;
```

```
                sum+=digit;
                key_var_digit=key_var_digit/10;
        }
        return(sum);
}
int length_calc(char stringlen[100])
{
        for(int len=0;stringlen[len]!='\0';++len);
        return(len);
}
int oddmeter(int key_var_digit2)
{
        int coin=0;
        if(key_var_digit2%2==0)
        {
                coin=0;
        }
        else if(key_var_digit2%2==1)
        {
                coin=1;
        }
        return(coin);
}
int licence_key(char key[])
{
        int gex=2;
        int validation=0;
        if(key[9]!='\0')
        {
                gex=1;
        }
        else if(key[9]=='\0')
        {
                int x;
                x=key[1];
                int temp_key1=digit_sqncr(x);
                x=key[2];
                int temp_key2=digit_sqncr(x);
                if(temp_key1==temp_key2)
                {
                        validation++;
                }
                temp_key1=key[0];
                temp_key2=key[3];
```

```
                x=temp_key1+temp_key2;
                int check=oddmeter(x);
                if(check==1)
                {
                        validation++;
                }
                x=key[5];
                temp_key1=digit_sqncr(x);
                x=key[8];
                temp_key2=digit_sqncr(x);
                if(temp_key1==temp_key2)
                {
                        validation++;
                }
                temp_key1=key[6];
                temp_key2=key[7];
                x=temp_key1+temp_key2;
                check=oddmeter(x);
                if(check==1)
                {
                        validation++;
                }
                if(validation==4)
                {
                        gex=0;
                }
                else
                {
                        gex=1;
                }
        }
        return(gex);
}
class server_security
{
        private:
        char password[40];
        char security_question[50];
        char security_ans[20];

        public:

        void getdata()
        {
```

```
                cout<<"\n Enter the Password : ";
                cin>>password;
                cout<<" Enter The Security Question : ";
                gets(security_question);
                cout<<" Enter the answer of Security Question : ";
                gets(security_ans);
        }
        void putdata()
        {       cout<<"\n The Password : ";
                cout<<password;
                cout<<"\n The Security Question : ";
                cout<<security_question;
                cout<<"\n Enter the answer of Security Question : ";
                cout<<security_ans;
        }
        void modify()
        {
                char mpassword[40]=" ";
                char msecurity_question[50]=" ";
                char msecurity_ans[20]=" ";
                std_err3();
                cout<<" Enter the Password : ";
                cin>>mpassword;
                std_err3();
                cout<<" Enter The Security Question : ";
                gets(msecurity_question);
                std_err3();
                cout<<" Enter the answer of Security Question : ";
                gets(msecurity_ans);
                if(strcmp(mpassword,"~")!=0)
                {
                        strcpy(password,mpassword);
                }
                if(strcmp(msecurity_question,"~")!=0)
                {
                        strcpy(security_question,msecurity_question);
                }
                if(strcmp(msecurity_ans,"~")!=0)
                {
                        strcpy(security_ans,msecurity_ans);
                }
        }
};
class server_login : public server_security
```

```
{
        private:
        char protocol[10];
        char site_name[15];
        char id[36];
        char domain[10];

        public:

        void getdata()
        {
                cout<<"\n Enter the Transfer Protocol : ";
                cin>>protocol;
                cout<<" Enter the Site Name : ";
                gets(site_name);
                cout<<" Enter the Domain  : ";
                cin>>domain;
                cout<<" Enter the Login ID ( complete ): ";
                gets(id);
                server_security :: getdata();
        }
        void putdata()
        {
                cout<<"\n The Transfer Protocol : ";
                cout<<protocol;
                cout<<"\n The Site Name : ";
                cout<<site_name;
                cout<<"\n The Domain  : ";
                cout<<domain;
                cout<<"\n Login ID : ";
                cout<<id;
                server_security :: putdata();
        }
        char* return_id()
        {
                return(id);
        }
        char* return_sitename()
        {
                return(site_name);
        }
        void modify()
        {
                char mprotocol[10]=" ";
```

```
                char msite_name[15]=" ";
                char mid[36]=" ";
                char mdomain[10]=" ";
                std_err3();
                cout<<" Enter the new Transfer Protocol : ";
                cin>>mprotocol;
                std_err3();
                cout<<" Enter the new Site Name : ";
                gets(msite_name);
                std_err3();
                cout<<" Enter the new Domain  : ";
                cin>>mdomain;
                std_err3();
                cout<<" Enter the new  Login ID ( complete ): ";
                gets(mid);
                if(strcmp(mprotocol,"~")!=0)
                {
                        strcpy(protocol,mprotocol);
                }
                if(strcmp(msite_name,"~")!=0)
                {
                        strcpy(site_name,msite_name);
                }
                if(strcmp(mdomain,"~")!=0)
                {
                        strcpy(domain,mdomain);
                }
                if(strcmp(mid,"~")!=0)
                {
                        strcpy(id,mid);
                }
                server_security :: modify();
        }
};
class user : public server_login
{
        private:

        char f_name[20];
        char l_name[20];

        public:

        void getdata()
```

```
	{
		cout<<"\n Enter The First Name of User : ";
		gets(f_name);
		cout<<" Enter The Last Name of the User : ";
		gets(l_name);
		server_login :: getdata();
	}
	void putdata()
	{
		cout<<"\n The First Name of the User is : ";
		cout<<f_name;
		cout<<"\n The Last Name of the User is : ";
		cout<<l_name;
		server_login :: putdata();
	}
	void modify()
	{
		char mf_name[20]=" ";
		char ml_name[20]=" ";
		std_err3();
		cout<<" Enter the  New First Name of User  : ";
		gets(mf_name);
		std_err3();
		cout<<" Enter the New Last Name of the User : ";
		gets(ml_name);
		if(strcmp(mf_name,"~")!=0)
		{
			strcpy(f_name,mf_name);
		}
		if(strcmp(ml_name,"~")!=0)
		{
			strcpy(l_name,ml_name);
		}
		server_login :: modify();
	}
};
int logix(int logix_low,unsigned char logix_in,int logix_high)
{
	unsigned int logix_in_conver=logix_in;
	if((logix_in_conver>=logix_low)&&(logix_in_conver<=logix_high))
	{
		return(0);
	}
	else
```

```
	{
		return(1);
	}
}
void encom_lgo()
{
	cout<<"\n\n\t\t\t"<<b<<b<<b<<b<<b<<b<<"  ";
	cout<<"\n\t\t\t"<<b<<"            ";
	cout<<"\n\t\t\t"<<b<<b<<b<<b<<"   "<<b<<b<<"  "<<b<<"  ";
	cout<<b<<b<<b<<b<<"  "<<b<<b<<b<<b<<b<<"   "<<b<<b<<" "<<b<<b;
	cout<<"\n\t\t\t"<<b<<"      "<<b<<" "<<b<<" "<<b<<"  ";
	cout<<b<<"   "<<"  "<<b<<"   "<<b<<"   "<<b<<" "<<b<<" "<<b;
	cout<<"\n\t\t\t"<<b<<b<<b<<b<<b<<b<<"  "<<b<<"  "<<b<<b<<"  ";
	cout<<b<<b<<b<<b<<"  "<<b<<b<<b<<b<<b<<"   "<<b<<"   "<<b;
}
void ln_drw(int a)
{
	if(a==1)
	{
		for(int i=0;i<80;++i)
		{
			cout<<"=";
		}
	}
	else
	{	for(int i=0;i<80;++i)
		{
			cout<<"_";
		}
	}

}
void mrgn_sqncr(int spc,int activate_line)
{
	for(int i=0;i<spc;++i)
	{
		cout<<"\t";
	}
	if(activate_line==0)
	{
		cout<<ll;
	}

}
```

```
int finder(char hexspace[50])
{
	int hext=length_calc(hexspace);
	int hexc=1;
	for(int hexb=0;hexb<hext;++hexb)
	{
		if(hexspace[hexb]==' ')
		{
			hexc=0;
			break;
		}
	}
	return(hexc);
}
void hedr_tle1()
{
	cout<<"\t\t\t    Your Smart Password Manager \n";
}
void hedr_tle2()
{
	cout<<"\n\t\t\t         Terminal Window\n";

}
void devlprs()
{
	cout<<"\n";
	cout<<(char)1;
	cout<<" Encom Vers 0.01 \n  Devloped by Harshit Yadav\n\n";
}
void fwrd_msg()
{
	cout<<"\n  PRESS ANY KEY TO CONTINUE :\>";
}
int dsp_gui1()
{
	unsigned char mnu_1;
	int mnu_flag=0;
	cout<<"\n";
	mrgn_sqncr(6,0);
	cout<<" Select An Choice to continue :";
	mrgn_sqncr(6,0);
	cout<<"  1. User Profile\n ";
	mrgn_sqncr(6,0);
	cout<<"  2. Save Credential\n";
```

```
        mrgn_sqncr(6,0);
        cout<<"  3. Manage Credential\n";
        mrgn_sqncr(6,0);
        cout<<"  4. FUN ZONE \n ";
        mrgn_sqncr(6,0);
        cout<<"  5. Control Panel\n ";
        mrgn_sqncr(6,0);
        cout<<"  6. About us\n ";
        mrgn_sqncr(6,0);
        cout<<"  0. Exit\n\n ";
        mrgn_sqncr(6,1);
        cout<<"  --> ";
        cin>>mnu_1;
        mnu_flag=logix(48,mnu_1,54);
        while(mnu_flag)
        {
                mrgn_sqncr(5,1);
                std_err2();
                cout<<"     --> ";
                cin>>mnu_1;
                mnu_flag=logix(48,mnu_1,54);

        }
        int rt_disp_guide=mnu_1;
        return(rt_disp_guide);

}
void login_time()
{
        time_t rawtime;
        struct tm * timeinfo;
        time (&rawtime);
        timeinfo = localtime (&rawtime);
        cout<<asctime(timeinfo);

}
void profile_creator()
{
        clrscr();
        ln_drw(1);
        mrgn_sqncr(2,2);
        cout<<"SIGN UP TERMINAL \n";
        ln_drw(1);
        char case_49_loop;
```

```
int case_49_loop_var1=0;
char app_name[30];
char hexalgo[30];
cout<<" Remember Any Field should not contain ' ' SPACE \n";
cout<<" Enter Your Profile Name          : ";
gets(app_name);
while(1)
{
        int oss=1;
        oss=finder(app_name);
        if(oss==0)
        {
                cout<<"\n ERROR \n";
                cout<<" Remember Any Field should not contain ' ' SPACE \n";
                cout<<" Enter Your Profile Name          : ";
                gets(app_name);
        }
        if(oss==1)
        {
                break;
        }
}
cout<<" Enter You Password              : ";
gets(hexalgo);
while(1)
{
        int oss=1;
        oss=finder(hexalgo);
        if(oss==0)
        {
                cout<<"\n ERROR \n";
                cout<<" Remember Any Field should not contain ' ' SPACE \n" ;
                cout<<" Enter You Password              : ";
                gets(hexalgo);
        }
        if(oss==1)
        {
                break;
        }
}
cout<<"\n \n Select From Below : ";
cout<<"\n 6. Exit ";
cout<<"\n 7. Submit ";
cout<<"\n 8. Retry \n-->  ";
```

```
	cin>>case_49_loop;
	case_49_loop_var1=logix(54,case_49_loop,56);
	while(case_49_loop_var1)
	{
		std_err2();
		cin>>case_49_loop;
		case_49_loop_var1=logix(54,case_49_loop,56);
	}
	switch(case_49_loop)
	{
		case 54:
		{
			exit(1);
		}
		case 55:
		{
			ofstream hex1("ADMIN.exe",ios::out);
			hex1<<app_name<<' '<<hexalgo<<' ';
			hex1.close();
			cout<<"\n Changes Saved Sucessfully Press any key to Exit and Apply
Changes...";
			getch();
			exit(1);
		}
		case 56:
		{
			profile_creator();
		}
	}
}
void bootstrap()
{
	clrscr();
	ln_drw(1);
	mrgn_sqncr(4,2);
	cout<<"Login Manager \n";
	ln_drw(1);
	cout<<"\n\n";
	cout<<"Select an option from below : \n";
	cout<<" 7. To Exit Application \n";
	cout<<" 8. Reset Your Profile : ";
	cout<<"\n --> ";
	int boot;
	cin>>boot;
```

```
        while(1)
        {
                if(boot==7)
                {
                        clrscr();
                        gotoxy(26,14);
                        cout<<" :> Event Sucessfully Completed <: ";
                        gotoxy(26,16);
                        cout<<" :> Shutting Down Application <: ";
                        delay(1000);
                        exit(1);
                }
                if(boot==8)
                {
                        clrscr();
                        cout<<"\n Enter APPLICATION KEY TO RESET YOUR PASSWORD ";
                        char passkey[9];
                        cout<<"\n (ie XXXX-XXXX )\n KEY == ";
                        cin>>passkey;
                        int bot=licence_key(passkey);
                        if(bot==0)
                        {
                                profile_creator();
                        }
                        else
                        {
                                clrscr();
                                gotoxy(26,14);
                                cout<<" :>    Invalid Key Entered     <: ";
                                gotoxy(26,16);
                                cout<<" :> Shutting Down Application <: ";
                                delay(1000);
                                exit(1);
                        }

                }

        }


}
void tron();
void password()
{
```

```
int value;
char arr[20],ch;
int j;
for(int i=0;;i++)
{
	gotoxy(26,14);
	cout<<"Enter your password : ";
	j=0;
	while((ch=getch())!=13)
	{
		cout<<"*";
		arr[j]=ch;
		j++;
	}
	arr[j]='\0';
	if(strcmp(arr,"ENCOM")==0)
	{
		bootstrap();
	}
	else
	{
		ifstream file_in("ADMIN.exe",ios::in);
		char arr1[50];
		char arr2[50];
		int scan=0;
		while(!file_in.eof())
		{
			if(scan==0)
			{
				file_in>>arr1;

			}
			if(scan==1)
			{
				file_in>>arr2;

				break;
			}
			scan++;
		}
		file_in.close();
		if(strcmp(arr2,arr)==0)
		{
			break;
```

```
                        }
                        else
                        {
                                gotoxy(33,12);
                                cout<<"INVALID PASSWORD ";
                                delay(700);
                                gotoxy(26,14);
                                cout<<" :> Restarting Application <: ";
                                delay(1000);
                                tron();

                        }
                }
        }

}
void tron()
{
        int tron_1=0;
        int tron_2=0;
        int tron_3=0;
        clrscr();
        cout<<"\n MESSAGE : Activating Tron (Runtime Application Gaurdian)";
        cout<<"\n\n\n\n\n";
        delay(1000);
        cout<<"\nMESSAGE : TRON STATUS REPORT::\n\n";
        cout<<"\n Diagnostics Complete at : \n\n\n\t\t\t";
        login_time();
        cout<<"\n";
        cout<<"\n  Request :   System File                                    [";
        ofstream file_check1;
        file_check1.open("EncomQ.exe",ios::nocreate);
        if(!file_check1)
        {
                cout<<" >__< ";
                tron_1=1;
        }
        else
        {
                cout<<" VALID ";
                tron_1=2;
        }
        cout<<"]\n";
        file_check1.close();
```

```
delay(500);
cout<<"  Request :      User File                                ";
ifstream file_check2;
file_check2.open("ADMIN.exe",ios::in,ios::nocreate);
if(!file_check2)
{
        cout<<" NOT FOUND ";
        tron_2=2;
}
else
{
        cout<<" FOUND ";
        tron_2=3;

}
file_check2.close();
cout<<"]\n";
delay(500);
cout<<"  Request :      Vault File                               ";
ofstream file_check3;
file_check3.open("vault.exe",ios::noreplace);
if(!file_check3)
{
        cout<<" FOUND ";
}
else
{
        cout<<" NOT FOUND ";
        tron_3=3;
}
cout<<"]\n";
file_check3.close();
fwrd_msg();
getch();
if(tron_1==1)
{
        clrscr();
        gotoxy(17,12);
        cout<<"ENCOM : System File Invaild Appliction Will Terminate ";
        gotoxy(28,17);
        fwrd_msg();
        getch();
        exit(1);
}
```

```
        if(tron_2==3)
        {
                clrscr();
                gotoxy(9,10);
                cout<<"! If you have forgot your Password or want to exit Type ENCOM ";
                password();
        }
        if(tron_2==2)
        {
                clrscr();
                gotoxy(1,12);
                cout<<"USER FILE NOT FOUND :CREATE NEW PROFILE ( Previous Record File will
be deleted )";
                gotoxy(22,15);
                cout<<"Press 1 To Confirm Else Press Any Key To Exit ";
                int rom;
                gotoxy(40,16);
                cin>>rom;
                if(rom==1)
                {
                        gotoxy(28,18);
                        fwrd_msg();
                        getch();
                   remove("vault.exe");
                        profile_creator();
                }
                else
                {
                        exit(0);
                }
        }
        if(tron_3==3)
        {
                clrscr();
                gotoxy(13,12);
                cout<<" APPLICATION FILE MISSING: Previous Record File Not Found  ";
                gotoxy(28,17);
                fwrd_msg();
                getch();
        }
}
int intelligence(char intellarr[])
{
        int intelsize=length_calc(intellarr);
```

```
	long int intlsum=0;
	if(stricmp(intellarr,"exit")==0)
	{
		return(0);
	}
	else
	{
		for(int intllloop=0;intllloop<intelsize;++intllloop)
		{
			intlsum+=intellarr[intllloop];
		}
		int intl_para;
		intl_para=oddmeter(intlsum);
		switch(intl_para)
		{
			case 0:
			{
				cout<<"\n Encom  : Yes ";
				break;
			}
			case 1:
			{
				cout<<"\n Encom  : No ";
				break;
			}
		}
		return(1);
	}
}
void main()
{
	clrscr();
	encom_lgo();
	cout<<"\n\n";
	ln_drw(2);
	hedr_tle1();
	devlprs();
	fwrd_msg();
	getch();
	clrscr();
	tron();
	char start='n';
	do
	{
```

```
clrscr();
tp_lgo();
cout<<"\n";
ln_drw(2);
int mnu_switch_guide=dsp_gui1();
switch(mnu_switch_guide)
{
        case 49:
        {
                clrscr();
                ln_drw(1);
                mrgn_sqncr(4,2);
                cout<<"USER PROFILE \n";
                ln_drw(1);
                ifstream file_in("ADMIN.exe",ios::in);
                char hexer[80];
                char arr1[50];
                char arr2[50];
                int scan=0;
                if(!file_in)
                {
                        cout<<"\n ERROR : ADMIN.exe NOT FOUND  ";
                        getch();
                        exit(1);
                }
                while(!file_in.eof())
                {
                        if(scan==0)
                        {
                                file_in>>arr1;
                                cout<<"\n Your Profile Name is     : "<<arr1<<"\n";
                        }
                        if(scan==1)
                        {
                                file_in>>arr2;
                                cout<<"\n Your Password  is       : "<<arr2<<"\n";
                                break;
                        }
                        scan++;
                }
                file_in.close();
                cout<<"\n \n Select From Below : ";
                cout<<"\n 7. Reset ";
                cout<<"\n 8. Main Menu \n--> ";
```

```
                char ul;
                cin>>ul;
                int ulc=logix(55,ul,56);
                while(ulc)
                {
                        std_err2();
                        cin>>ul;
                        ulc=logix(55,ul,56);
                }
                switch(ul)
                {
                        case 55:
                        {
                                clrscr();
                                profile_creator();
                                break;

                        }
                        case 56:
                        {
                                break;
                        }
                }
                break;
        }
        case 50:
        {
                clrscr();
                user alpha;
                ln_drw(1);
                mrgn_sqncr(4,2);
                cout<<"ENTER YOUR DETAILS \n";
                ln_drw(1);
                cout<<"\n Object Created of size : ";
                cout<<sizeof(alpha);
                cout<<" Bytes ";
                alpha.getdata();
                getch();
                cout<<" \n\n ";
                cout<<"\n Writing into the memory ";
                ofstream fout;
                fout.open("vault.exe",ios::out|ios::app|ios::binary);
                fout.write((char*)&alpha,sizeof(alpha));
                fout.close();
```

```
                cout<<"\n Record Sucessfully Writen into File ";
                getch();
                break;

        }
        case 51:
        {
                clrscr();
                ln_drw(1);
                mrgn_sqncr(4,2);
                cout<<"RETRIVING DATA \n";
                ln_drw(1);
                cout<<"\n\nRecords : ";
                user retrive[6];
                int ocr=0;
                ifstream retro;
                retro.open("vault.exe",ios::binary|ios::in|ios::nocreate);
                if(!retro)
                {
                        cout<<"\n No data in File to Display ! ";
                        getch();
                        break;

                }
                for(int r=1;!retro.eof();++r)
                {

                        retro.read((char*)&retrive[r],sizeof(user));
                        if(!retro.eof()==0)
                        {
                                break;
                        }
                        ocr=1;
                        cout<<"\n"<<" "<<r<<". "<<retrive[r].return_id()<<"    (
"<<retrive[r].return_sitename()<<" )";
                        if(r==5)
                        {
                                cout<<"\n Want to Retrive more Data (y/n) ";
                                char sdk;
                                cin>>sdk;
                                if(sdk=='y')
                                {
                                        r=1;
                                        clrscr();
```

```
                                        }
                                        if(sdk=='n')
                                        {
                                                break;
                                        }
                                }

                        }
                        if(ocr==1)
                        {
                                retro.close();
                                cout<<"\n\n Choose The Site Number whose data you want
retrive : ";
                                r-=1;
                                int sdk_no;
                                cin>>sdk_no;
                                while(1)
                                {
                                        if(sdk_no<=r)
                                        {
                                                break;
                                        }
                                        else
                                        {
                                                std_err2();
                                                cin>>sdk_no;
                                        }
                                }
                                clrscr();
                                ln_drw(1);
                                mrgn_sqncr(4,2);
                                cout<<retrive[sdk_no].return_sitename()<<"\n";
                                ln_drw(1);
                                retrive[sdk_no].putdata();
                                cout<<"\n \n Select From Below : ";
                                cout<<"\n 7. Edit ";
                                cout<<"\n 8. Delete  ";
                                cout<<"\n 9. Main Menu \n--> ";
                                char retroo;
                                cin>>retroo;
                                int retrool=logix(55,retroo,57);
                                while(retrool)
                                {
                                        std_err2();
```

```
                                        cin>>retroo;
                                        retrool=logix(55,retroo,57);
                                }
                                switch(retroo)
                                {
                                        case 55:
                                        {
                                                clrscr();
                                                ln_drw(1);
                                                mrgn_sqncr(4,2);
                                                cout<<"EDIT OPTION \n";
                                                ln_drw(1);
                                                user retrive_e;
                                                fstream
retr("vault.exe",ios::in|ios::out|ios::binary);
                                                long rpos=0;
                                                while(!retr.eof())
                                                {
                                                        rpos=retr.tellg();

        retr.read((char*)&retrive_e,sizeof(user));
                                                        {

        if((strcmp(retrive_e.return_id(),retrive[sdk_no].return_id())==0)&&(strcmp(retrive_e.retu
rn_sitename(),retrive[sdk_no].return_sitename())==0));
                                                                {

                                                                        retrive_e.modify();
                                                                        cout<<"\n File
Modified ";
                                                                        retr.seekg(rpos);

        retr.write((char*)&retrive_e,sizeof(user));
                                                                        cout<<"\n File
Written ";
                                                                        break;
                                                                }
                                                        }

                                                }
                                                retr.close();
                                                cout<<"\n Message : File Sucessfully Written
";
                                                break;
```

```
                                        }
                                        case 56:
                                        {
                                                user retrive_e;
                                                clrscr();
                                                ln_drw(1);
                                                mrgn_sqncr(4,2);
                                                cout<<"DELETE OPTION \n";
                                                ln_drw(1);
                                                char del='n';
                                                fstream
finc6("vault.exe",ios::in|ios::out|ios::binary);
                                                fstream
finc7("temp.exe",ios::in|ios::out|ios::binary);
                                                while(!finc6.eof())
                                                {

        finc6.read((char*)&retrive_e,sizeof(user));
                                                        if(!finc6.eof()==0)
                                                        {
                                                                break;
                                                        }

        if(strcmp(retrive_e.return_id(),retrive[sdk_no].return_id())==0)
                                                        {
                                                                retrive_e.putdata();
                                                                cout<<"\n Are You Sure You
Want to Delete Data (y/n) : ";
                                                                cin>>del;
                                                                if(del=='n')
                                                                {

        finc7.write((char*)&retrive_e,sizeof(user));
                                                                }
                                                                if(del=='y')
                                                                {
                                                                        cout<<"\n Record
Sucessfully Deleted : ";
                                                                }
                                                        }

                                                        else
                                                        {
```

```
finc7.write((char*)&retrive_e,sizeof(user));
                                                }

                                                }
                                                finc6.close();
                                                finc7.close();
                                                remove("vault.exe");
                                                rename("temp.exe","vault.exe");
                                                break;
                                        }
                                        case 57:
                                        {
                                                break;
                                        }
                                }
                                cout<<"\n Press any key to continue ";
                                getch();
                        }
                        else
                        {
                                cout<<"\n\n File Empty Nothing To Display ! ";
                        }
                        break;
                }
                case 52:
                {
                        clrscr();
                        int intl_loopask=1;
                        ln_drw(1);
                        mrgn_sqncr(4,2);
                        cout<<"TIME PASS \n";
                        ln_drw(1);
                        cout<<"\n MESSAGE : All the Reply By The System will be Yes Or No
:";
                        cout<<"\n {Type exit/EXIT to exit this Menu}";
                        while(intl_loopask==1)
                        {
                                cout<<"\n Encom  : Enter the Question you want ask  ";
                                cout<<"\n User   : ";
                                char intlques[100];
                                gets(intlques);
                                intl_loopask=intelligence(intlques);
                        }
```

```
                        getch();
                        break;

                }
                case 53:
                {
                        clrscr();
                        tp_lgo();
                        cout<<"\n";
                        ln_drw(1);
                        mrgn_sqncr(4,2);
                        cout<<"CONTROL PANEL \n";
                        ln_drw(1);
                        cout<<"\n \n Select From Below : ";
                        cout<<"\n 6. Corrupt APPLICATION ";
                        cout<<"\n 7. Delete User Profile ( Saved Data associated with it will
also be deleted ) ";
                        cout<<"\n 8. Delete File Containing Saved File  ";
                        cout<<"\n 9. Exit \n -->";
                        int sos=0;
                        cin>>sos;
                        if(sos==9)
                        {
                                break;
                        }
                        cout<<"\n Enter APPLICATION KEY TO CARRY OUT ANY OF THE
FUNCTION ";
                        char appkey[9];
                        cout<<"\n (ie XXXX-XXXX )\n KEY == ";
                        cin>>appkey;
                        int h=licence_key(appkey);
                        if(h==0)
                        {
                                getch();
                                clrscr();
                                switch(sos)
                                {
                                        case 6:
                                        {
                                                remove("EncomQ.exe");
                                                gotoxy(26,14);
                                                cout<<" :> Event Sucessfully Completed <: ";
                                                gotoxy(26,16);
                                                cout<<" :> Shutting Down Application <: ";
```

```
                                        delay(1000);
                                        exit(1);
                                }
                                case 7:
                                {
                                        remove("ADMIN.exe");
                                        gotoxy(26,14);
                                        cout<<" :> Event Sucessfully Completed <: ";
                                        gotoxy(26,16);
                                        cout<<" :> Shutting Down Application <: ";
                                        delay(1000);
                                        exit(1);

                                }
                                case 8:
                                {
                                        remove("vault.exe");
                                        gotoxy(26,14);
                                        cout<<" :> Event Sucessfully Completed <: ";
                                        gotoxy(26,16);
                                        cout<<" :> Shutting Down Application <: ";
                                        delay(1000);
                                        exit(1);
                                }
                                default:
                                {
                                        gotoxy(26,14);
                                        cout<<" :> Event Failed (Invalid Parameter)
<: ";
                                        gotoxy(26,16);
                                        cout<<" :> Shutting Down Application <: ";
                                        delay(1000);
                                        exit(1);
                                }
                        }
                }
                else
                {

                cout<<"Invalid Key ";
                }
                break;

        }
```

```
                        case 54:
                        {
                                clrscr();
                                ln_drw(1);
                                mrgn_sqncr(5,2);
                                cout<<"ABOUT \n";
                                ln_drw(1);
                                encom_lgo();
                                cout<<"\n\n";
                                hedr_tle1();
                                devlprs();
                                cout<<"  Email ID : harshityadav95@gmail.com ";
                                cout<<"\n  Visit : http:\\actroidnotex.blogspot.in \n\n";
                                fwrd_msg();
                                getch();
                                break;
                        }
                        case 48:
                        {
                                exit(1);
                        }

                }
                cout<<"\n\n\nExit To Main Menu (y/n) : ";
                cin>>start;
        }while(start=='y');
        getch();
}
```