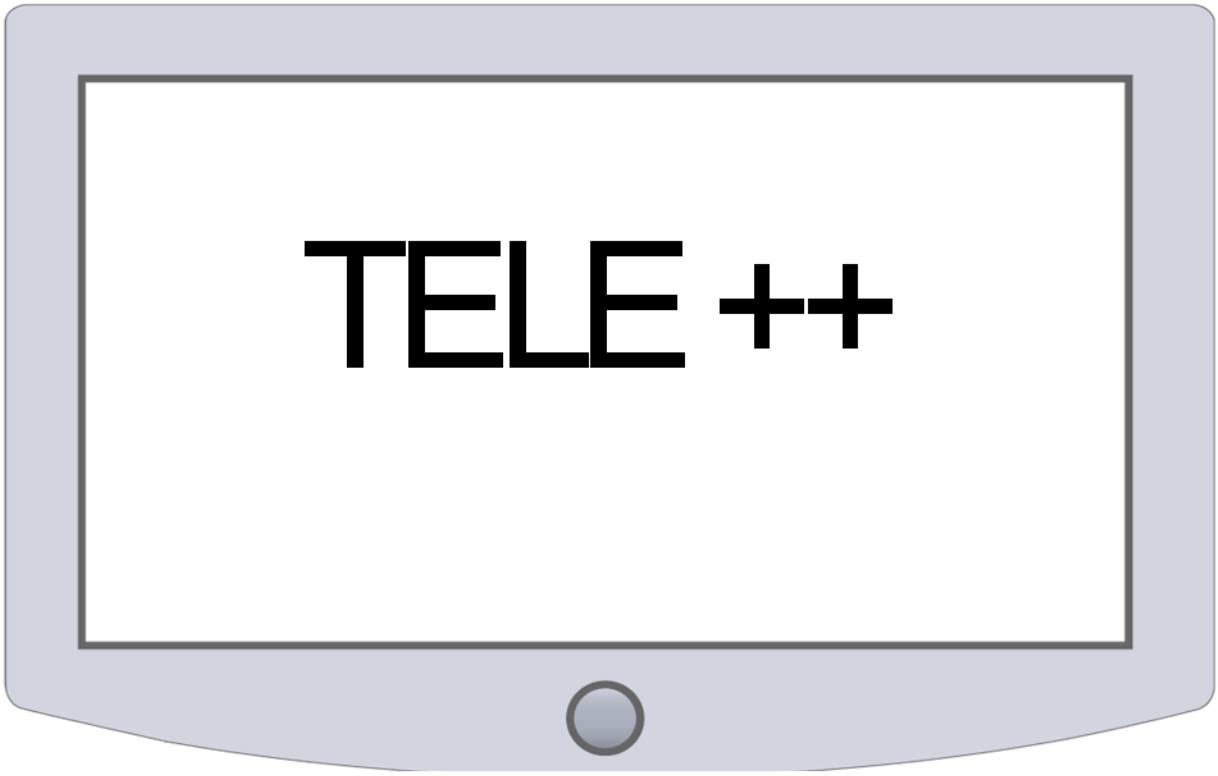
CS PROJECT



A Project by,

Rohith.V &

M.Hariprasad.

(2013-2014)

COMPUTER SCIENCE DEPARTMENT

BONAFIDE CERTIFICATE

This is to certify that the project work on "**Tele ++ – Direct to computer**" with reference to Computer Science Department of The Hindu Senior Secondary School is a bonafide work carried out by **ROHITH.V** and **HARI PRASAD.M** , students of class **XII-A**.

Submitted for Practical examination held on _____ at **THE HINDU SENIOR SECONDARY SCHOOL, CHENNAI-20**.

_____ _____

Internal Examiner External Examiner

ACKNOWLEDGEMENT

We express our sincere thanks to our Computer Science teacher **Mrs. DHANALAKSHMI MANIVASAGAM, B.Sc., M.C.A.,** without whose guidance and unstinting help, this project would never have been a success.

We would like to thank our families for their undying support and source of encouragement and my friends for being a source of inspiration.

Last, but definitely not the least, I thank my Principal **Smt.Padmini Sriraman** and my Vice-Principals **Sri.A.Kasi Viswanathan** , **Smt.S.Rajakalai** and the School Management for providing us with excellent laboratory facilities.

INTRODUCTION

TELE++ is a modest effort in creating a program for TV aficionados who want to see TV anywhere and everywhere. Our research has shown that lifestyles are getting busier in today's day and age. Everyday people spend hours in transit or waiting. Today, over 60 percent of the population use internet to view videos on their smart phones and tablets either at home or at office.

TELE++ is a unique service for all those valued subscribers who want to keep in touch with their favourite shows, programmes and channels that they have subscribed on TELE++ even when they are on move through their C++ DOS.

CLASSES AND FUNCTION DESCRIPTION

Class User:

To get details of user and display it in neat format.

Class Channel:

To get channel description and display those it in a neat format.

Void user :: input()

Gets the user details and generates USER ID.

Void user :: output()

Prints details of the user in a neat format.

Void user::controls()

To get choice from the user and perform the operation chosen by the user from the list of user operations.

Void user :: check()

To check whether the user's balance has reached the state of critical balance range corresponding to the pack selected.

Char user :: retrpk()

Returns the pack name selected by the user.

Char* user :: retrname()

Returns the user name.

Int user :: retrid()

Returns the user id.

Void user :: smaldisp()

Prints the user id and the current balance in the account.

Int user::retrbalance()

Returns the balance.

Void main_user()

To make any amendments in the user's account and his/her particulars.

Void channel::indata()

To update a new channel and give the programme schedule of the corresponding channel.

Void channel :: output()

Displays a channel's programme schedule along with program information.

Void channel :: operations()

To perform channel operations.

Void channel :: smalldp()

To display channel number and channel name.

Void channel :: seebyuser()

Prints the channel's name, number and programme schedule.

Char channel :: retpack()

Returns the pack.

Int channel :: retno()

Returns the channel number.

Char* channel :: retname()

Returns channel name.

Void main_channel()

To make amendments in the channel data.

Void feed_out()

Prints the feedback given by the user which is written into a text file.

Void feedback_in(int idin, char idin2[50])

To write feedback about the service into a text file.

Void ads_in()

To create advertisements for channels pertaining to specific packs.

Void menu_chanint(int , char)

To select the channel which the user wants to view from the list of channels available by entering the channel number.

Void main_chanint(char pk1)

Checks whether the data file is readable or not.

Void main_admin()

To perform administrative tasks of the service by the administrators.

Void menu_uslogin(int)

For the user to choose the desired option once he/she logs into his account.

Void main_us login()

To login to the user’s account by entering user ID and user name.

Int main()

Displays home screen and gets user’s choice to perform the kind of function or to get the kind information they want to.

ALGORITHM

START

1. Print Home Screen
2. Declare integer i.
3. Read i.
4. 1.If i==1

    4.1.1.Call main_uslogin()

    4.2.Else if i==2

    4.2.1.Call main_admin()

    4.3.Else if i==3

    4.3.1.Print Credits

    4.4.Else

    4.4.1.End Program by calling exit(0).

STOP

CODE

```
#include<fstream.h>
#include<conio.h>
#include<iomanip.h>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>


/* To implement offline TeleVision Service */
//Definition of class "user"

class user{ protected:
                char name[20];
                int id;
                char address[4][30];
                char phone[20];
                char pack;
                int balance;

                public:
                user();
                int retrid();
                char retrpk();
```

```
        int retrbalance();
        char* retrname();
        void detbal(int);
        void input();
        void output();
        void controls();
        void smaldisp();
        void check();
    };

user listu1, list_tmpu;

 int sn, an, mn;

user::user()
{   strcpy(name," ");
    int i;
    for(i=0;i<4;i++)
        strcpy(address[i]," ");
    strcpy(phone," ");
    pack='n';
    balance=0;
    id=0;
}
```

```
void user::detbal(int v)
{       balance-=v;
}


void user::input()
{       ifstream textfilein1("NUM-U.txt",ios::in);
int c=1;
char string1[20];
        if(!textfilein1)
                cout<<"Error in reading num-u.txt"<<endl;
        else
        {       while(textfilein1>>string1)
                {       if(c==1)
                                sn=atoi(string1);
                        else if(c==2)
                                an=atoi(string1);
                        else if(c==3)
                                mn=atoi(string1);
                        ++c;
                }
                textfilein1.close();
        }
```

```
cout<<"Enter Name"<<endl;
gets(name);
cout<<"Enter Address"<<endl;
int i;
for(i=0;i<4;i++)
        gets(address[i]);
cout<<"Enter Phone Number"<<endl;
gets(phone);
cout<<"Enter Pack Type :"<<endl;
cout<<"Saver(s) / Annual (a) / Mega (m)"<<endl;
cin>>pack;
while(pack!='a' && pack!='s' && pack!='m')
{       cout<<"Enter Pack Type Again"<<endl;
        cin>>pack;
}
switch(pack)
{       case 's' : cout<<"Opening Balance >=1000"<<endl;
                        break;
        case 'a' : cout<<"Opening Balance >=2000"<<endl;
                        break;
        default : cout<<"Opening Balance >=3000"<<endl;
                        break;
}
cout<<"Enter Opening Balance"<<endl;
```

```
cin>>balance;
while(pack=='s' && balance<1000)
{	cout<<"Enter Balance Again"<<endl;
	cin>>balance;
}
while(pack=='a' && balance<2000)
{	cout<<"Enter Balance Again"<<endl;
	cin>>balance;
}
while(pack=='m' && balance<3000)
{	cout<<"Enter Balance Again"<<endl;
	cin>>balance;
}
switch(pack)
{	case 's' : id=sn;
			break;
	case 'a' : id=an;
			break;
	default : id=3*mn;
			break;
}
cout<<"User ID : \t"<<id<<endl;
++sn, ++an, ++mn;
```

```
}


void user::output()
{       cout<<"Identiy Number : \t"<<id<<endl;
        cout<<"Name : \t"<<name<<endl;
        cout<<"Address : ";
        int i;
        for(i=0;i<4;i++)
                cout<<'\t'<<address[i]<<endl;
        cout<<"Phone Number : \t"<<phone<<endl;
        cout<<"Balance : \t"<<balance<<endl;
        cout<<"Pack :\t";
        switch(pack)
        {       case 's': cout<<"Saver"<<endl;
                          break;
                case 'a': cout<<"Annual"<<endl;
                          break;
                default : cout<<"Mega"<<endl;
                          break;
        }
}
```

```
void user::controls()
{       int c4u=10;
        while(c4u==10)
        {       cout<<"User Name : \t"<<name<<endl;
                cout<<"User ID : \t"<<id<<endl;
                cout<<"User Operations"<<endl;
                cout<<"1.Change Address"<<endl;
                cout<<"2.Change Phone Number"<<endl;
                cout<<"3.Change Pack"<<endl;
                cout<<"4.Update Balance"<<endl;
                cout<<"5.Exit"<<endl;
                cout<<"Enter Your Choice"<<endl;
                cin>>c4u;
                if(c4u==1)
                {       for(c4u=0;c4u<4;c4u++)
                                strcpy(address[c4u]," ");
                        cout<<"Enter New Address"<<endl;
                        for(c4u=0;c4u<4;c4u++)
                                gets(address[c4u]);
                }
                else if(c4u==2)
                {       strcpy(phone," ");
                        cout<<"Enter New Phone Number"<<endl;
                        gets(phone);
```

```
}
else if(c4u==3)
{	cout<<"Enter New Pack Type :"<<endl;
	cout<<"Saver(s) / Annual (a) / Mega (m)"<<endl;
	cin>>pack;
	while(pack!='a' && pack!='s' && pack!='m')
	{	cout<<"Enter Pack Type Again"<<endl;
		cin>>pack;
	}
}
else if(c4u==4)
{	cout<<"Enter Amount Paid"<<endl;
	cin>>c4u;
	balance+=c4u;
	while(pack=='s' && balance<1000)
	{	cout<<"Enter Balance Again"<<endl;
		cin>>c4u;
		balance+=c4u;
	}
	while(pack=='a' && balance<2000)
	{	cout<<"Enter Balance Again"<<endl;
		cin>>c4u;
		balance+=c4u;
	}
```

```
                while(pack=='m' && balance<3000)
                {       cout<<"Enter Balance Again"<<endl;
                        cin>>c4u;
                        balance+=c4u;
                }
        }
        else if(c4u==5)
                return;
        else
                cout<<"Wrong Choice"<<endl;
        cout<<"Press 1 to continue User Operations"<<endl;
        cin>>c4u;
        if(c4u==1)
                c4u=10;
        else
                c4u=0;
    }
}


void user::check()
{       int c5=0, flag=0;
        while(pack=='s' && balance<1000)
        {       cout<<"Balance less than Critical Balance"<<endl;
```

```
        cout<<"Enter Balance Again"<<endl;
        cin>>c5;
        balance+=c5;
        flag=1;
}
while(pack=='a' && balance<2000)
{       cout<<"Balance less than Critical Balance"<<endl;
        cout<<"Enter Balance Again"<<endl;
        cin>>c5;
        balance+=c5;
        flag=1;
}
while(pack=='m' && balance<3000)
{       cout<<"Balance less than Critical Balance"<<endl;
        cout<<"Enter Balance Again"<<endl;
        cin>>c5;
        balance+=c5;
        flag=1;
}
if(flag==0)
{       cout<<"Balance is fine"<<endl;
        cout<<"CurrentPrints Balance : "<<balance<<endl;
        cout<<"User ID : "<<id<<endl;
        cout<<"User Name : "<<name<<endl;
```

```
        }
}


char user::retrpk()
{       return pack;
}


char* user::retrname()
{       return name;
}


int user::retrid()
{       return id;
}


void user::smaldisp()
{  cout<<"Welcome , "<<name<<" !!"<<endl;
        cout<<"User ID : \t"<<id<<endl;
        cout<<"A/C Balance : \t"<<balance<<endl;
}
```

```
int user::retrbalance()
{       return balance;
}

//Member Functions of User are linked below in main_user()

void main_user()
{       int c1u=10, tmpuid=0;
                ifstream textfilein1("NUM-U.txt",ios::in);
int c=1;
char stringu[50];
char string1[20];
        if(!textfilein1)
                cout<<"Error in reading num-u.txt"<<endl;
        else
        {       while(textfilein1>>string1)
                {       if(c==1)
                                sn=atoi(string1);
                        else if(c==2)
                                an=atoi(string1);
                        else if(c==3)
                                mn=atoi(string1);
```

```
            ++c;
        }
        textfilein1.close();
}
while(c1u==10)
{  clrscr();
        cout<<"User Controls Menu"<<endl;
        cout<<"1.See all users.dat"<<endl;
        cout<<"2.See a particular user"<<endl;
        cout<<"3.Join a user"<<endl;
        cout<<"4.Check for Balance"<<endl;
        cout<<"5.User Operations"<<endl;
        cout<<"6.Exit"<<endl;
        cout<<"Enter Choice"<<endl;
        cin>>c1u;
        if(c1u==1)
        {       ifstream fileuin1("users.dat",ios::in|ios::binary);
                if(!fileuin1)
                        cout<<"Error in reading users.dat"<<endl;
                else
                {       while(fileuin1.read((char*)&listu1,sizeof(listu1)))
                        {       listu1.output();
                                getch();
                                cout<<endl<<endl<<endl;
```

```
                }
                fileuin1.close();
        }
}
else if(c1u==2)
{       ifstream fileuin2("users.dat",ios::in|ios::binary);
        if(!fileuin2)
                cout<<"Error in reading users.dat"<<endl;
        else
        {  cout<<"Enter User ID"<<endl;
                cin>>tmpuid;
                cout<<"Enter User Name"<<endl;
                gets(stringu);
                while(fileuin2.read((char*)&listu1,sizeof(listu1)))
                {       if(tmpuid==listu1.retrid() &&
strcmpi(stringu,listu1.retrname())==0)
                        {       listu1.output();
                                tmpuid=-1;
                                break;
                        }
                }
                if(tmpuid!=-1)
                        cout<<"User Not Found"<<endl;
                fileuin2.close();
        }
```

```
            }
            else if(c1u==3)
            {       ofstream
fileuout1("users.dat",ios::out|ios::app|ios::binary);
                    if(!fileuout1)
                            cout<<"Error in writing users.dat"<<endl;
                    else
                    {       list_tmpu.input();
                            fileuout1.write((char*)&list_tmpu,sizeof(list_tmpu));
                            fileuout1.close();
                    }
            }
            else if(c1u==4 )
            {  ifstream fileuin3("users.dat",ios::in|ios::binary);
                    ofstream
fileutmp1("tusers1.dat",ios::out|ios::app|ios::binary);
                    if(!fileuin3 || !fileutmp1)
                            cout<<"Error in reading users.dat or writing
tusers1.dat"<<endl;
                    else
                    {       while(fileuin3.read((char*)&listu1,sizeof(listu1)))
                            {       listu1.check();
                                    fileutmp1.write((char*)&listu1,sizeof(listu1));
                            }
                            fileuin3.close();
```

```
                        fileutmp1.close();
                        if(!remove("users.dat"))
                                cout<<"Old File Removed"<<endl;
                        if(!rename("tusers1.dat","users.dat"))
                                cout<<"New File Renamed"<<endl;
                }
        }
        else if(c1u==5)
        {       ifstream fileuin4("users.dat",ios::in|ios::binary);
                ofstream
fileutmp2("tusers2.dat",ios::out|ios::app|ios::binary);
                if(!fileuin4 || !fileutmp2)
                        cout<<"Error in reading users.dat or writing
tusers2.dat";
                else
                {       while(fileuin4.read((char*)&listu1,sizeof(listu1)))
                        {       listu1.check();
                                fileutmp2.write((char*)&listu1,sizeof(listu1));
                        }
                        fileuin4.close();
                        fileutmp2.close();
                        ifstream fileuin5("users.dat",ios::in|ios::binary);
                        ofstream
fileutmp3("tusers3.dat",ios::out|ios::app|ios::binary);
                        if(!fileuin5 || !fileutmp3)
```

```
                cout<<"Error in reading users.dat  or writng
tusers3.dat"<<endl;
            else
            {   cout<<"Enter User ID"<<endl;
                cin>>tmpuid;

    while(fileuin5.read((char*)&listu1,sizeof(listu1)))
                {   if(tmpuid==listu1.retrid())
                    {   listu1.controls();
                        tmpuid=-1;
                    }
                    if(tmpuid!=-1)
                        cout<<"User Not Found"<<endl;

    fileutmp3.write((char*)&listu1,sizeof(listu1));
                }
                if(tmpuid!=-1)
                    cout<<"User Not Found"<<endl;
                fileuin5.close();
                fileutmp3.close();
                if(!remove("users.dat"))
                    cout<<"Old File Removed"<<endl;
                if(!rename("tusers3.dat","users.dat"))
                    cout<<"New File Renamed"<<endl;
            }
```

```
            }
        }
        else if(c1u==6)
            return;
        else
            cout<<"Wrong Choice"<<endl;
        cout<<"Press 1 to continue User Main Menu"<<endl;
        cin>>c1u;
        if(c1u==1)
            c1u=10;
        else
            c1u=0;
}
ofstream textfileuout("NUM-U.txt",ios::out);
if(!textfileuout)
        cout<<"Error in writing num-u.txt"<<endl;
else
{       textfileuout<<sn;
        textfileuout.put(' ');
        textfileuout<<an;
        textfileuout.put(' ');
        textfileuout<<mn;
        textfileuout.put(' ');
}
```

```
        textfileuout.close();
        getch();
}


//Definition of class "channel"



class channel{  protected:
                int no;
                char name[20];
                char program[50][50];
                char info[50][100];
                char startingtime[50][10];
                char endingtime[50][10];
                int count;
                char pack;

                public:
                channel();
                char retpack();
                char* retname();
                int retno();
```

```
            void smalldp();
            void seebyuser();
            void indata();
            void outdata();
            void operations();
        };

int saverno, annualno, megano;


channel list_c;

channel::channel()
{  strcpy(name," ");
       for(int i=0;i<50;i++)
       {      strcpy(program[i]," ");
              strcpy(startingtime[i]," ");
              strcpy(endingtime[i]," ");
              strcpy(info[i]," ");
       }
       count=0;
       no=0;
       pack='n';
}
```

```
void channel::indata()
{       strcpy(name," ");
        for(int i=0;i<50;i++)
        {       strcpy(program[i]," ");
                strcpy(startingtime[i]," ");
                strcpy(endingtime[i]," ");
                strcpy(info[i]," ");
        }
        count=0;
        no=0;
        pack='n';
        cout<<"Enter Channel Name"<<endl;
        gets(name);
        int c2=1, tmp;
        cout<<endl<<endl;
        cout<<"Enter Programme Schedule"<<endl;
        while(c2<=50)
        {       cout<<endl;
                cout<<"Enter Starting Time"<<endl;
                gets(startingtime[count]);
                while(c2>1 && strcmpi(startingtime[count],endingtime[count-
1])!=0)
```

```
		{	cout<<"No Continuity in Time Limit !!"<<endl;
			cout<<"Enter Starting Time Again"<<endl;
			gets(startingtime[count]);
		}
		cout<<"Enter Programme Name"<<endl;
		gets(program[count]);
		cout<<"Enter Information about the Programme (" ", if no information)"<<endl;
		gets(info[count]);
		if(strcmpi(info[count]," ")==0)
			strcpy(info[count],"No Information Available");
		cout<<"Enter Ending Time"<<endl;
		gets(endingtime[count]);
		++count;
		cout<<"Press 1 to add Schedule "<<endl;
		cin>>tmp;
		if(tmp!=1)
			break;
		++c2;
	}
	while(strcmpi(startingtime[0],endingtime[count-1])!=0 && count!=1)
	{	cout<<"No Continuity in Time Limit"<<endl;
		cout<<"Enter Ending Time again"<<endl;
		gets(endingtime[count-1]);
	}
```

```
        cout<<endl<<endl;
        cout<<"Enter Pack Type"<<endl;
        cout<<"Saver(s) / Annual(a) / Mega(m)"<<endl;
        cin>>pack;
        while(pack!='s' && pack!='a' && pack!='m')
        {       cout<<endl<<"Enter Pack Type Again"<<endl;
                cin>>pack;
        }
        switch(pack)
        {       case 's': no=saverno;
                          ++saverno;
                          break;
                case 'a': no=annualno;
                          ++annualno;
                          break;
                default: no=megano;
                          ++megano;
                        break;
        }
}



void channel::outdata()
```

```
{       cout<<endl<<endl;
        cout<<"Name : "<<name<<endl;
        cout<<"Programme Schedule : "<<endl;
        for(int i=0;i<count;i++)
        {       cout<<startingtime[i]<<" - "<<endingtime[i]<<" :
"<<program[i]<<endl;
                cout<<'\t'<<'\t'<<"(\t"<<info[i]<<"\t)"<<endl;
                cout<<endl;
        }
        cout<<"Pack : "<<'\t';
        switch(pack)
        {       case 's': cout<<"Saver"<<endl;
                          break;
                case 'a': cout<<"Annual"<<endl;
                          break;
                default: cout<<"Mega"<<endl;
                        break;
        }
        cout<<"Number : \t"<<no<<endl;
}



void channel::operations()
{       int c2=10, c3, tmp;
```

```
char pck;
while(c2==10)
{       cout<<"Channel : \t"<<name<<endl;
        cout<<"Channel Operations"<<endl;
        cout<<"1.Change Schedule of a Channel"<<endl;
        cout<<"2.Change Pack Type of a Channel"<<endl;
        cout<<"3.Exit"<<endl;
        cout<<"Enter Your Choice"<<endl;
        cin>>c2;
        if(c2==1)
        {       cout<<"Enter New Programme Schedule"<<endl;
                for(c3=0;c3<50;c3++)
                {       strcpy(startingtime[c3]," ");
                        strcpy(endingtime[c3]," ");
                        strcpy(info[c3]," ");
                        strcpy(program[c3]," ");
                }
                c3=1;
                count=0;
                while(c2<=50)
                {       cout<<endl;
                        cout<<"Enter Starting Time"<<endl;
                        gets(startingtime[count]);
                        while(c2>1 &&
strcmpi(startingtime[count],endingtime[count-1])!=0)
```

```
                {       cout<<"No Continuity in Time Limit !!"<<endl;
                        cout<<"Enter Starting Time Again"<<endl;
                        gets(startingtime[count]);
                }
                cout<<"Enter Programme Name"<<endl;
                gets(program[count]);
                cout<<"Enter Information about the Programme("
",if no information)"<<endl;
                gets(info[count]);
                if(strcmpi(info[count]," ")==0)
                        strcpy(info[count],"No Information Available");
                cout<<"Enter Ending Time"<<endl;
                gets(endingtime[count]);
                ++count;
                cout<<"Press 1 to add Schedule "<<endl;
                cin>>tmp;
                if(tmp!=1)
                        break;
                ++c2;
        }
        while(strcmpi(startingtime[0],endingtime[count-1])!=0)
        {       cout<<"No Continuity in Time Limit !!"<<endl;
                cout<<"Enter Ending Time Again"<<endl;
                gets(endingtime[count-1]);
        }
```

```
}
else if(c2==2)
{       pck=pack;
        cout<<"Enter New Pack Type"<<endl;
        cout<<"Saver(s) / Annual(a) / Mega(m)"<<endl;
        cin>>pack;
        while(pack!='s' && pack!='a' && pack!='m')
        {       cout<<endl<<"Enter Pack Type Again"<<endl;
                cin>>pack;
        }
        switch(pck)
        {       case 's' : --saverno;
                           break;
                case 'a' : --annualno;
                           break;
                default  : --megano;
                           break;
        }
        switch(pack)
        {       case 's' : no=saverno;
                           ++saverno;
                           break;
                case 'a' : no=annualno;
                           ++annualno;
```

```
                                        break;
                        default  : no=megano;
                                        ++megano;
                                        break;
                        }
                }
                else if(c2==3)
                        return;
                else
                        cout<<"Wrong Choice"<<endl;
                cout<<"Press 1 to continue Channel Operations"<<endl;
                cin>>c2;
                if(c2==1)
                        c2=10;
                else
                        c2=0;
        }
}


void channel::smalldp()
{       cout<<no<<'\t'<<name<<endl;
}
```

```
void channel::seebyuser()
{	cout<<no<<'\t'<<name<<endl<<endl;
	for(int i=0;i<count;i++)
	{	cout<<startingtime[i]<<" - "<<endingtime[i]<<" :
"<<program[i]<<endl;
		cout<<'\t'<<'\t'<<"(\t"<<info[i]<<"\t)"<<endl;
		cout<<endl;
	}
}


char channel::retpack()
{	return pack;
}


int channel::retno()
{	return no;
}


char* channel::retname()
{	return name;
}
```

```
//main_channel() links all member functions of class "channel"



void main_channel()
{       char string[20];
        int c=1;
        ifstream textfilein("NUM-C.txt",ios::in);
        if(!textfilein)
                cout<<"Error in reading num-c.txt"<<endl;
        else
        {       while(textfilein>>string)
                {       if(c==1)
                                saverno=atoi(string);
                        else if(c==2)
                                annualno=atoi(string);
                        else if(c==3)
                                megano=atoi(string);
                        ++c;
                }
                textfilein.close();
        }
```

```
        c=1;
        while(c!=0)
        {       clrscr();
                cout<<"Channel Menu"<<endl;
                cout<<"1.Add Channel"<<endl;
                cout<<"2.Display Contents of channels.dat"<<endl;
                cout<<"3.Channel Settings"<<endl;
                cout<<"4.Exit"<<endl;
                cout<<"Enter Choice"<<endl;
                cin>>c;
                if(c==1)
                {       ofstream
fileout("channels.dat",ios::binary|ios::out|ios::app);
                        if(!fileout)
                                cout<<"Error in writing channels.dat"<<endl;
                        else
                        {       list_c.indata();
                                fileout.write((char*)&list_c,sizeof(list_c));
                        }
                        fileout.close();
                }
                else if(c==2)
                {       ifstream filein("channels.dat",ios::in|ios::binary);
                        if(!filein)
```

```
                cout<<"Error in reading channels.dat"<<endl;
        else
        {       while(filein.read((char*)&list_c,sizeof(list_c)))
                {       list_c.outdata();
                        getch();
                        cout<<endl<<endl<<endl;
                }
        }
        filein.close();
}
else if(c==3)
{       ifstream filein("channels.dat",ios::in|ios::binary);
        ofstream fileout2("t-channels.dat",ios::out|ios::binary);
        cout<<"Enter Channel Name"<<endl;
        gets(string);
        if(!filein)
                cout<<"Error in reading channels.dat"<<endl;
        else if(!fileout2)
                cout<<"Error in writing t-channels.dat"<<endl;
        else
        {       while(filein.read((char*)&list_c,sizeof(list_c)))
                {       if(strcmpi(list_c.retname(),string)==0)
                        {       c=30;
                                list_c.operations();
```

```
fileout2.write((char*)&list_c,sizeof(list_c));
                        }
                        else

fileout2.write((char*)&list_c,sizeof(list_c));
                }
                filein.close();
                fileout2.close();
                if(c==3)
                        cout<<"Channel Not Found"<<endl;
                if(!remove("channels.dat"))
                        cout<<"Old File Removed"<<endl;
                if(!rename("t-channels.dat","channels.dat"))
                        cout<<"Temporary File Renamed"<<endl;
        }
}
else if(c==4)
{       clrscr();
        cout<<"Good Bye!!"<<endl;
        getch();
 break;
}
else
        cout<<"Wrong Choice"<<endl;
```

```
            cout<<"Enter 1 to Continue Channel Menu"<<endl;
            cin>>c;
            clrscr();
        }
        ofstream textfileout("NUM-C.txt",ios::out);
        if(!textfileout)
            cout<<"Error in writing num-c.txt"<<endl;
        else
        {   textfileout<<saverno;
            textfileout.put(' ');
            textfileout<<annualno;
            textfileout.put(' ');
            textfileout<<megano;
            textfileout.put(' ');
        }
        textfileout.close();
        getch();



}



ifstream txtfeout("FEED.txt",ios::in);
```

```
char w2[50];

void feed_out()
{       if(!txtfeout)
                cout<<"Error in reading feed.txt"<<endl;
        else
        {       while(txtfeout>>w2)
                {       if(strcmpi(w2,"~")!=0)
                                cout<<w2<<endl;
                }
                txtfeout.close();
        }
        ofstream txt1("FEED.TXT",ios::out);
        if(!txt1)
                cout<<"Error in writing feed.txt"<<endl;
        else
        {       strcpy(w2," ");
                txt1<<w2;
                txt1.close();
        }
}

void feedback_in(int idin,char idin2[50])
```

```
{       char charfeed='!';
        clrscr();
        ofstream textfilefeed("FEED.txt",ios::out|ios::app);
        if(!textfilefeed)
                cout<<"Error in reading feed.txt"<<endl;
        else
        {       cout<<endl<<endl;
                cout<<"Enter Feed Back"<<endl;
                cout<<"CAUTION : Press ~ to stop writing feed back"<<endl;
                textfilefeed<<idin<<'\n'<<idin2<<'\n';
                while(charfeed!='~')
                {       cin.get(charfeed);
                        textfilefeed.put(charfeed);
                }
                char w[5];
                strcpy(w," ~ ");
                textfilefeed<<w;
                textfilefeed.close();
                getch();
        }
}
```

```
void ads_in()
{       int cad=100;
        clrscr();
        char charads='@';
        while(cad==100)
        {       cout<<endl<<endl;
                char q[5];
        clrscr();
                cout<<"Advertisements Menu"<<endl;
                cout<<"Create Advertisements for"<<endl;
                cout<<"1.Saver Pack"<<endl;
                cout<<"2.Annual Pack"<<endl;
                cout<<"3.Mega Pack"<<endl;
                cout<<"4.Exit"<<endl;
                cout<<"CAUTION : To stop writing ads , enter ~"<<endl;
                cout<<"Enter Choice"<<endl;
                cin>>cad;
                if(cad==1)
                {       ofstream textfileads1("SAV-A.txt",ios::out);
                        if(!textfileads1)
```

```
                cout<<"Error in reading sav-a.txt"<<endl;
        else
        {       cout<<"Enter Advertisement"<<endl;
                while(charads!='~')
                {       cin.get(charads);
                        textfileads1.put(charads);
                }
                strcpy(q," ~ ");
                textfileads1<<q;

        textfileads1.close();
        }
    }
    else if(cad==2)
    {       ofstream textfileads2("ANN-A.txt",ios::out);
            if(!textfileads2)
                cout<<"Error in reading ann-a.txt"<<endl;
            else
            {       cout<<"Enter Advertisement"<<endl;
                    while(charads!='~')
                    {       cin.get(charads);
                            textfileads2.put(charads);
                    }
strcpy(q," ~ ");
```

```
                textfileads2<<q;


        textfileads2.close();
        }
    }
    else if(cad==3)
    {   ofstream textfileads3("MEG-A.txt",ios::out);
        if(!textfileads3)
            cout<<"Error in reading meg-a.txt"<<endl;
        else
        {   cout<<"Enter Advertisement"<<endl;
            while(charads!='~')
            {   cin.get(charads);

textfileads3.put(charads);   }
            strcpy(q," ~ ");
            textfileads3<<q;
     textfileads3.put('~');


        textfileads3.close();  }
    }
    else if(cad==4)
        return;
    else
        cout<<"Wrong Choice"<<endl;
```

```
            cout<<"Enter 1 to Continue Advertisements Menu"<<endl;
            cin>>cad;
            if(cad==1)
                  cad=100;
            getch();
      }
}

void ads_out(char pk)
{     char word[500];
      if(pk=='s')
      {     ifstream tf1("SAV-A.txt",ios::in);
            if(!tf1)
                  cout<<"Error in reading sav-a.txt"<<endl;
            else
            {     while(tf1.getline(word,500,'~'))
                  {  cout<<"^^";
                        puts(word);
                        cout<<endl;
                  }
                  tf1.close();
            }
      }
      else if(pk=='a')
```

```
{       ifstream tf2("ANN-A.txt",ios::in);
        if(!tf2)
                cout<<"Error in reading ann-a.txt"<<endl;
        else
        {       while(tf2.getline(word,500,'~'))
                {  cout<<"^^";
                        puts(word);
                        cout<<endl;
                }
                tf2.close();
        }
}
else if(pk=='m')
{       ifstream tf3("MEG-A.txt",ios::in);
        if(!tf3)
                cout<<"Error in reading meg-a.txt"<<endl;
        else
        {       while(tf3.getline(word,500,'~'))
                {  cout<<"^^";
                        puts(word);
                        cout<<endl;
                }
                tf3.close();
        }
```

```
        }
}



ifstream filein7("channels.dat",ios::in|ios::binary);
channel inchanin;

int fchin=1;

void menu_chanint(int chano,char pk)
{       if(!filein7)
                cout<<"Error in reading channels.dat"<<endl;
        else
        {
        while(chano==200)
        {       cout<<"Channels on air..."<<endl;
                while(filein7.read((char*)&inchanin,sizeof(inchanin)))
                {       if(inchanin.retpack()==pk)
                                inchanin.smalldp();
                }
                filein7.clear();
                filein7.seekg(0,ios::beg);
                cout<<"Enter Channel Number to view  that channel "<<endl;
```

```
            cin>>chano;
            while(filein7.read((char*)&inchanin,sizeof(inchanin)))
            {       if(inchanin.retpack()==pk)
                    {       clrscr();
                            inchanin.seebyuser();
                            cout<<"Press any key to see channel menu"<<endl;
                            getch();
                            fchin=-1;
                            menu_chanint(200,inchanin.retpack());
                    }
            }
            if(fchin!=-1)
                    cout<<"Channel Number Unavailable"<<endl;
            return;
      }
      }
}
```

user inqw;

```
void main_chanint(char pk1, char nw[50])
{   ifstream fiu("users.dat",ios::in|ios::binary);
        clrscr();
        if(!filein7)
                cout<<"Error in reading channels.dat"<<endl;
        else
        {       menu_chanint(200,pk1);
                ofstream fil8("tusers.dat",ios::out|ios::app|ios::binary);
                if(!fil8)
                        cout<<"Error in writing tusers.dat or reading
users.dat"<<endl;
                else
                {       while(fiu.read((char*)&inqw,sizeof(inqw)))
                        {       if(strcmpi(inqw.retrname(),nw)==0)
                                {       if(pk1=='s')
                                                inqw.detbal(1);
                                        else if(pk1=='a')
                                                inqw.detbal(2);
                                        else if(pk1=='m')
                                                inqw.detbal(3);
                                        else
                                                cout<<"Wrong Pack"<<endl;
                                        fil8.write((char*)&inqw,sizeof(inqw));
                                }
```

```
                        fil8.write((char*)&inqw,sizeof(inqw));
                }
                if(!remove("users.dat"))
                        cout<<"Old File Removed"<<endl;
                if(!rename("tusers.dat","users.dat"))
                        cout<<"Temporary File Renamed"<<endl;
        }
}

        getch();
}

void main_admin()
{       int cadmin=300;
        clrscr();
        char p[10];
        cout<<"Enter Password : "<<endl;
        for(int i=0;i<8;i++)
        {       p[i]=getch();
                cout<<"*";
        }
        p[8]='\0';
        if(strcmp(p,"pass@123")!=0)
        {       cout<<endl<<"Wrong Password !!"<<endl;
```

```
        getch();
        return;
}
else{
while(cadmin==300)
        {       clrscr();
                cout<<"Welcome, Administrator !!"<<endl;
                cout<<"Tasks to perform : "<<endl;
                cout<<"1. Modify User Controls"<<endl;
                cout<<"2. Modify Channel Controls"<<endl;
                cout<<"3. Modify Advertisement Controls"<<endl;
                cout<<"4. View Feedback"<<endl;
                cout<<"5. Exit"<<endl;
                cout<<"Enter Choice"<<endl;
                cin>>cadmin;
                if(cadmin==1)
                        main_user();
                else if(cadmin==2)
                        main_channel();
                else if(cadmin==3)
                        ads_in();
                else if(cadmin==4)
                        feed_out();
                else if(cadmin==5)
```

```
                return;
        else
        {       cout<<"Wrong Choice"<<endl;
                cadmin=300;
        }
        cout<<"Enter 1 for Continuing Admin Home"<<endl;
        cin>>cadmin;
        if(cadmin==1)
                cadmin=300;
        else
                return;
    }   }
}

ifstream fileuin5("users.dat",ios::in|ios::binary);

user inuser;



void menu_uslogin(int cmin)
{       while(cmin==100)
        {       clrscr();
                inuser.smaldisp();
                cout<<endl<<endl;
```

```
int idlogus;
cout<<"ADVERITISEMENTS : "<<endl;
ads_out(inuser.retrpk()) ;
getch();
cout<<endl<<endl<<"1. View TV"<<endl;
cout<<"2. Give Feed Back"<<endl;
cout<<"3. Log Out"<<endl;
cout<<"Enter Choice"<<endl;
cin>>idlogus;
if(idlogus==1)
{       main_chanint(inuser.retrpk(),inuser.retrname());
        cmin=100;
}
else if(idlogus==2)
{       feedback_in(inuser.retrid(),inuser.retrname());
        cmin=100;
}
else if(idlogus==3)
{       cmin=100;
        return;
}
else
        cout<<"Wrong Choice"<<endl;
cout<<"Enter 1 for Continuing Home"<<endl;
```

```
            cin>>cmin;
            if(cmin==1)
                  menu_uslogin(100);
            else
                  return;
      }
}


void main_uslogin()
{     clrscr();
      char strlogus[50];
      int idlogus;
      if(!fileuin5 )
            cout<<"Error in reading data"<<endl;
      else
      {     cout<<"Enter User ID"<<endl;
            cin>>idlogus;
            cout<<"Enter User Name"<<endl;
            gets(strlogus);
            while(fileuin5.read((char*)&inuser,sizeof(inuser)))
            {     if(idlogus==inuser.retrid() &&
strcmpi(strlogus,inuser.retrname())==0)
                  {     idlogus=-1;
                        break;
```

```
                }
        }
        if(idlogus!=-1)
                cout<<"User Not Found"<<endl;
        else if(idlogus==-1)
        {       idlogus=100;
                menu_uslogin(idlogus);
        }
        fileuin5.close();
    }
    getch();
}



int main()
{ int i=6;
  while(i==6)
  {     clrscr();
  cout<<endl<<endl;
  cout<<"        TTTTTTT  EEEEEEE  LLL    EEEEEEE    ++     ++\n"
        <<"          TTT    EEE     LLL    EEE       ++     ++\n"
        <<"          TTT    EEEEEEE  LLL    EEEEEEE  ++++++++
++++++++\n"
        <<"          TTT    EEE     LLL    EEE       ++     ++\n"
```

```
            <<"          TTT    EEEEEEE   LLLLLLL  EEEEEEE     ++
++\n"
            <<"                                                  "
            <<"\n\n\n\n\n\n"
            <<setw(50)<<"                          D I R E C T     \n\n"
            <<setw(55)<<"                    T O          \n\n"
            <<setw(54)<<"              C O M P U T E R     \n\n\n"  ;
            getch();


            clrscr();
  cout<<"            _/\\_
_/\\_";cout<<endl<<endl<<endl;
  cout<<"        _/\\_
_/\\_";cout<<endl<<endl<<endl;
  cout<<setw(48)<<"  WE  WELCOME  YOU  TO "<<'\n'<<'\n';
  cout<<setw(51)<<"                      C++ abled SERVICES
"<<'\n'<<'\n';
  cout<<setw(47)<<"WITH  OUR GREETINGS"<<'\n';
  cout<<endl<<endl<<endl<<"         _/\\_
_/\\_";cout<<endl<<endl<<endl;
  cout<<"            _/\\_                          _/\\_";cout<<endl<<endl;
  cout<<"Press any key to Continue \n";
             getch();
             clrscr();
  cout<<"                         *  *\n"
              "                         *  /\\  *\n"
```

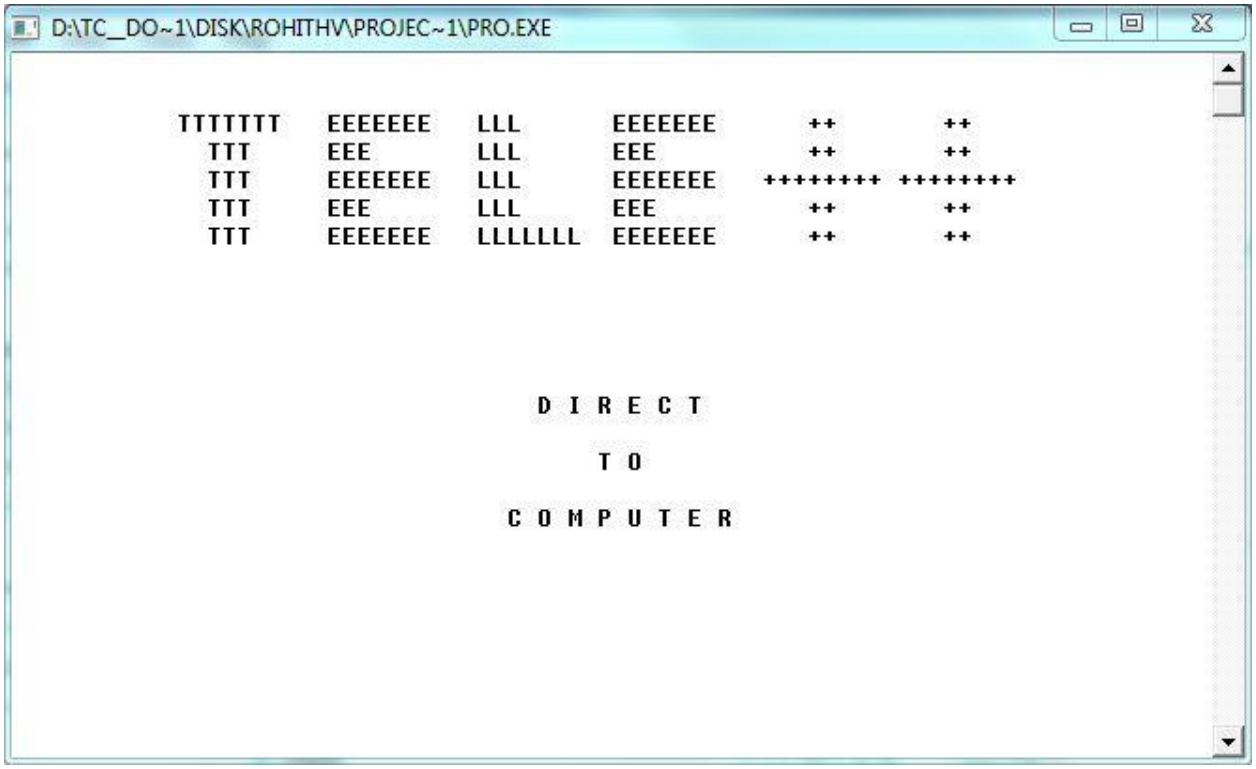
```
                "                    *   /  \\   *\n"
                "                   *    \\ /    *\n"
                "                   *    / \\    *\n"
                "                    * /    \\  *\n"
                "                      *      *\n"
                "                        *  * " ;
                getch();
                clrscr();
  cout <<'\n'<<'\n'<<'\n'<<'\n'<<"_/\\_                    1. U S E R  L O G I
N                      _/\\_\n"
                <<" T
T\n\n"
                <<" E                    2. A D M I N I S T R A T O R
E\n\n"
                <<" L
L\n\n"
                <<" E                     3. C R E D I T S
E\n\n"
                <<" +
+\n\n"
                <<" +                     4. E X I T
+\n\n"
                <<"_/\\_
_/\\_\n\n";


  cout<<"ENTER YOUR CHOICE : ";
  cin>>i;
```

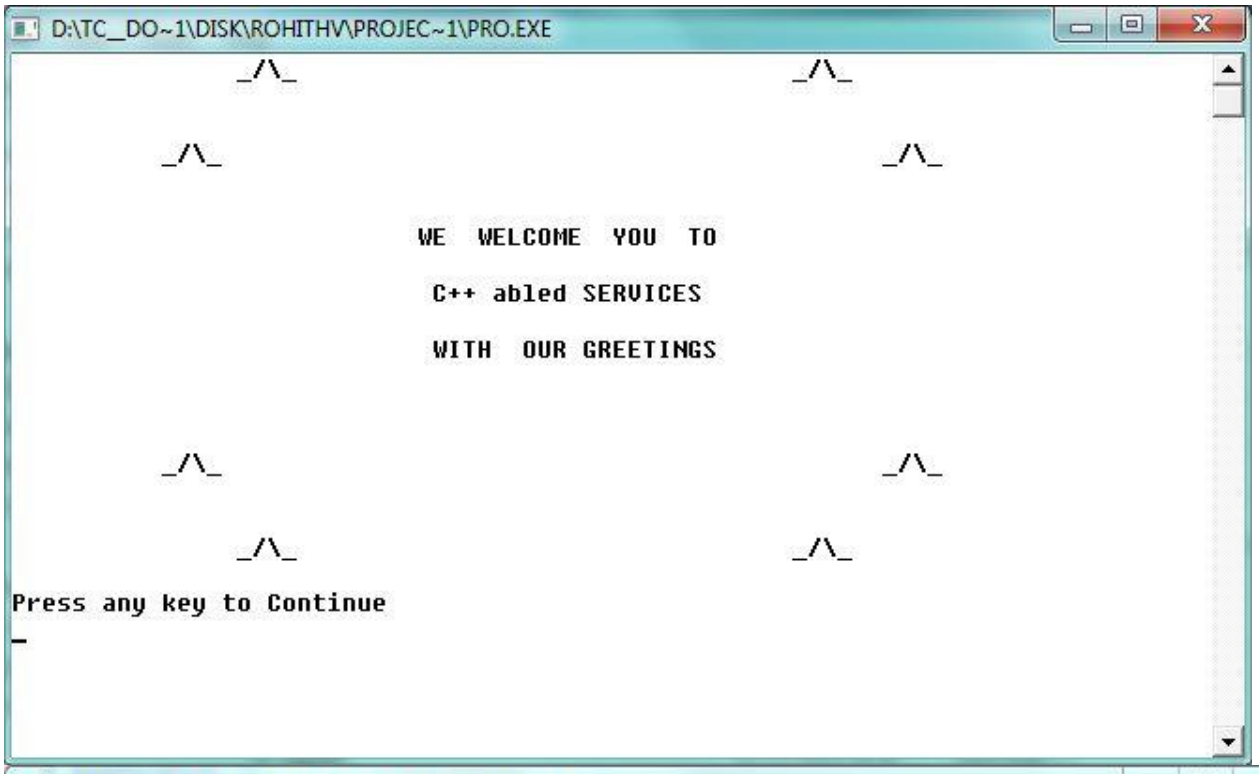
```
  if(i==1)
  {     main_uslogin();
                i=6;
  }
  else if(i==2)
  {     main_admin();
        i=6;
  }
  else if(i==3)
  {     clrscr();
        cout<<"CREDITS : "<<endl;
        cout<<"\t 1. TATA SKY for inspiring us with this idea. "<<endl;
        cout<<"\t 2. ROHITH.V and HARIPRASAD.M for effectively completing
this herculanean project."<<endl;
        cout<<"\t 3. Our COMPUTER SCINECE TEACHER, Mrs.
DHANALAKSHMI for guiding us throughout this project."<<endl;
        cout<<"\t 4. All the Channels for giving their programme
schedules."<<endl;
        getch();
        i=6;
  }
  else if(i==4)
  {     clrscr();
        cout<<"Take Care...\nBye\n"<<endl;
        getch();
```

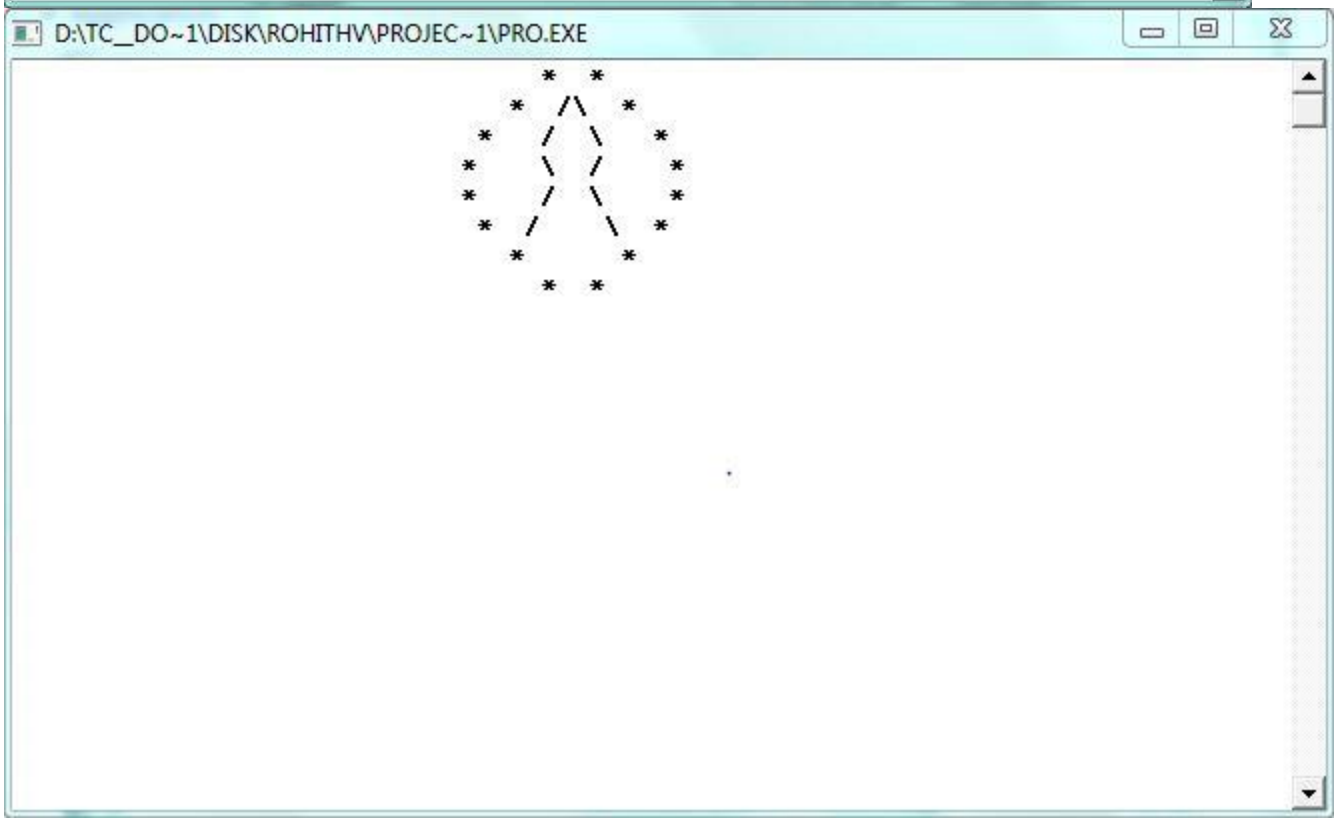
```
        i=6;
        exit(0);
  }
  else
  i=6;
  }
  return 0;
}
```

VIDEOS AND SNAPSHOTS

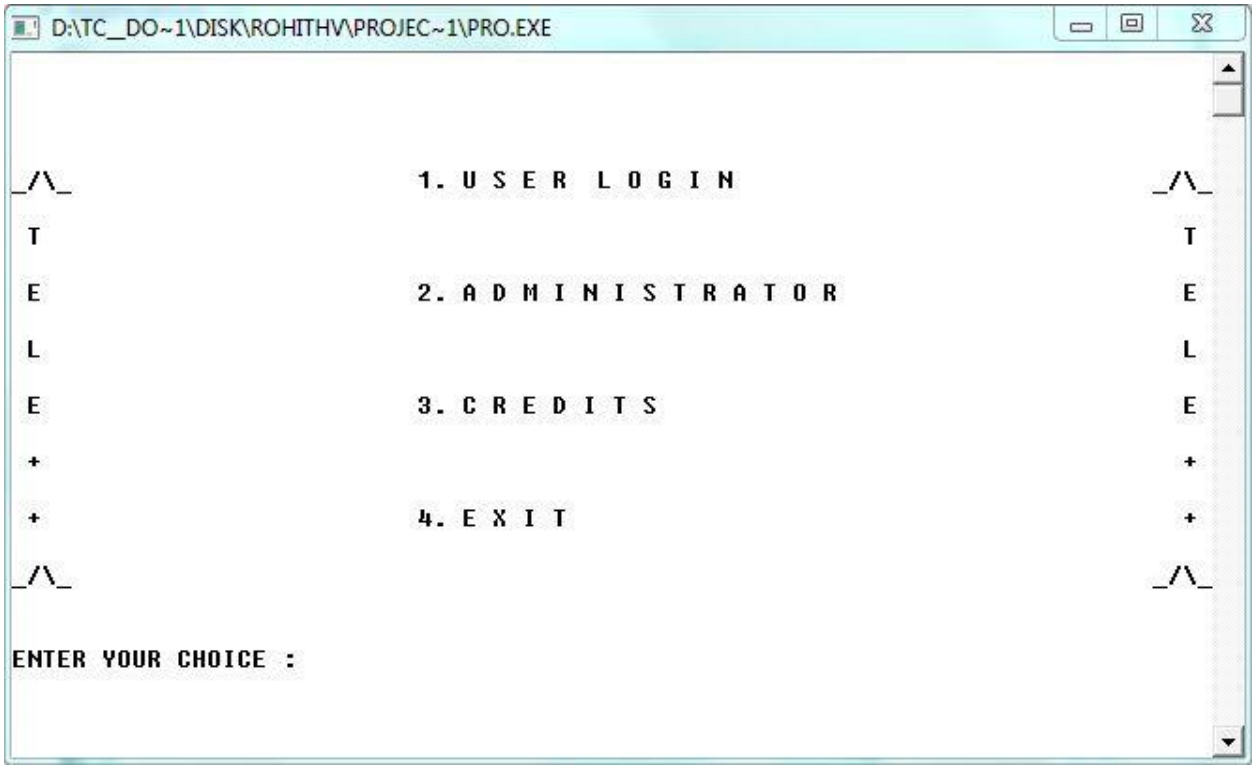
Home Screen



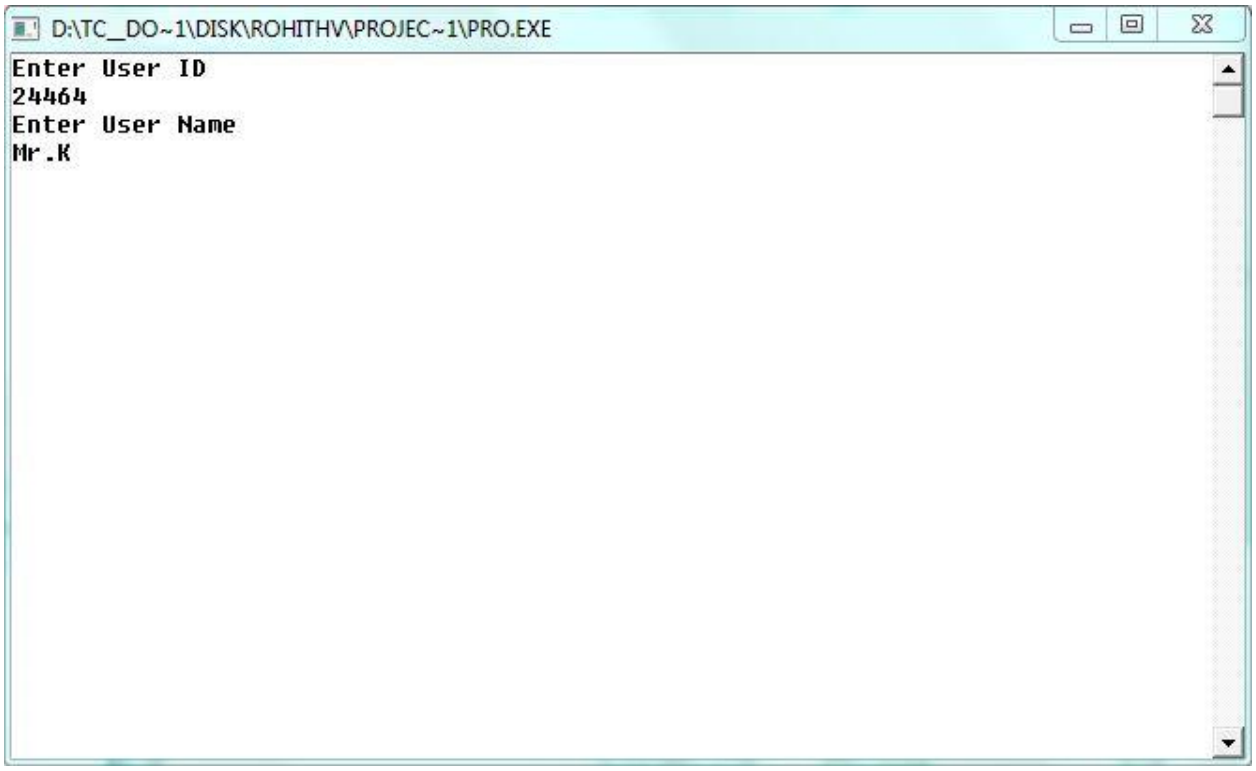

D:\TC__DO~1\DISK\ROHITHV\PROJEC~1\PRO.EXE
_/_
_/_
_/_
_/_
WE WELCOME YOU TO
C++ abled SERVICES
WITH OUR GREETINGS
_/_
_/_
_/_
_/_
Press any key to Continue


D:\TC__DO~1\DISK\ROHITHV\PROJEC~1\PRO.EXE

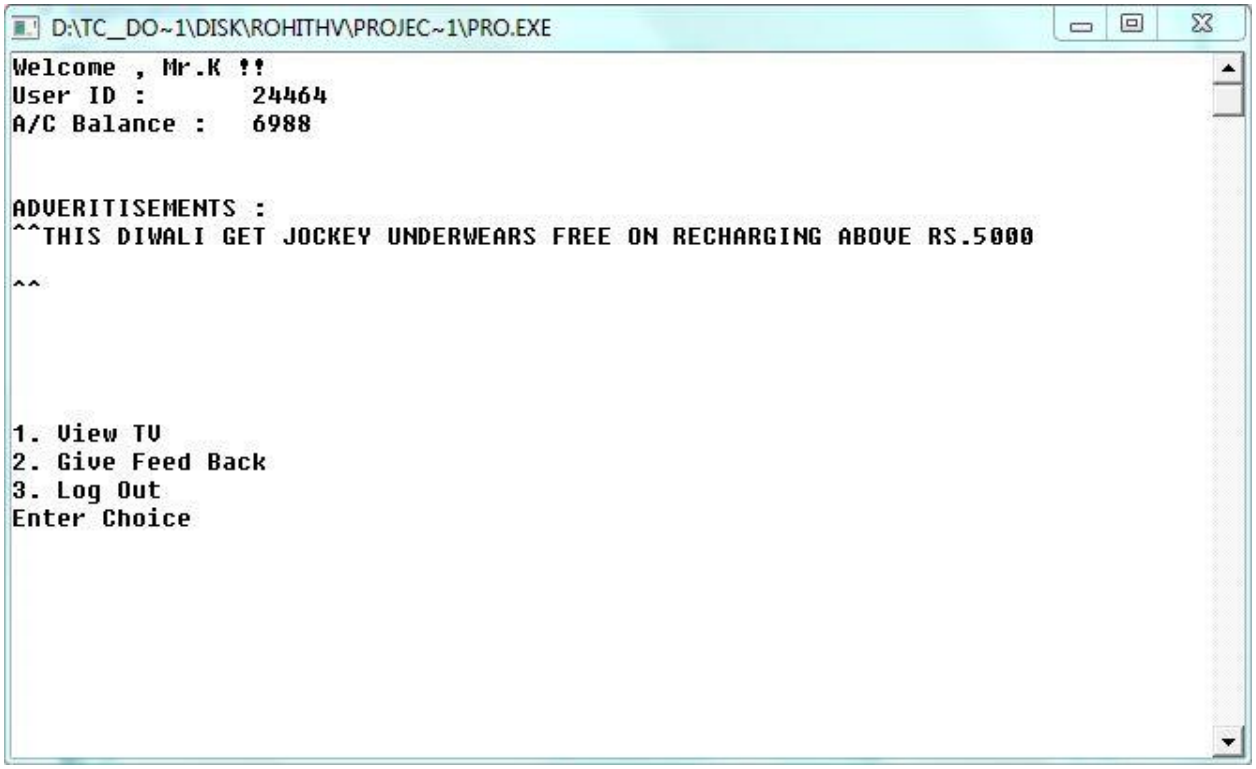
Menu:



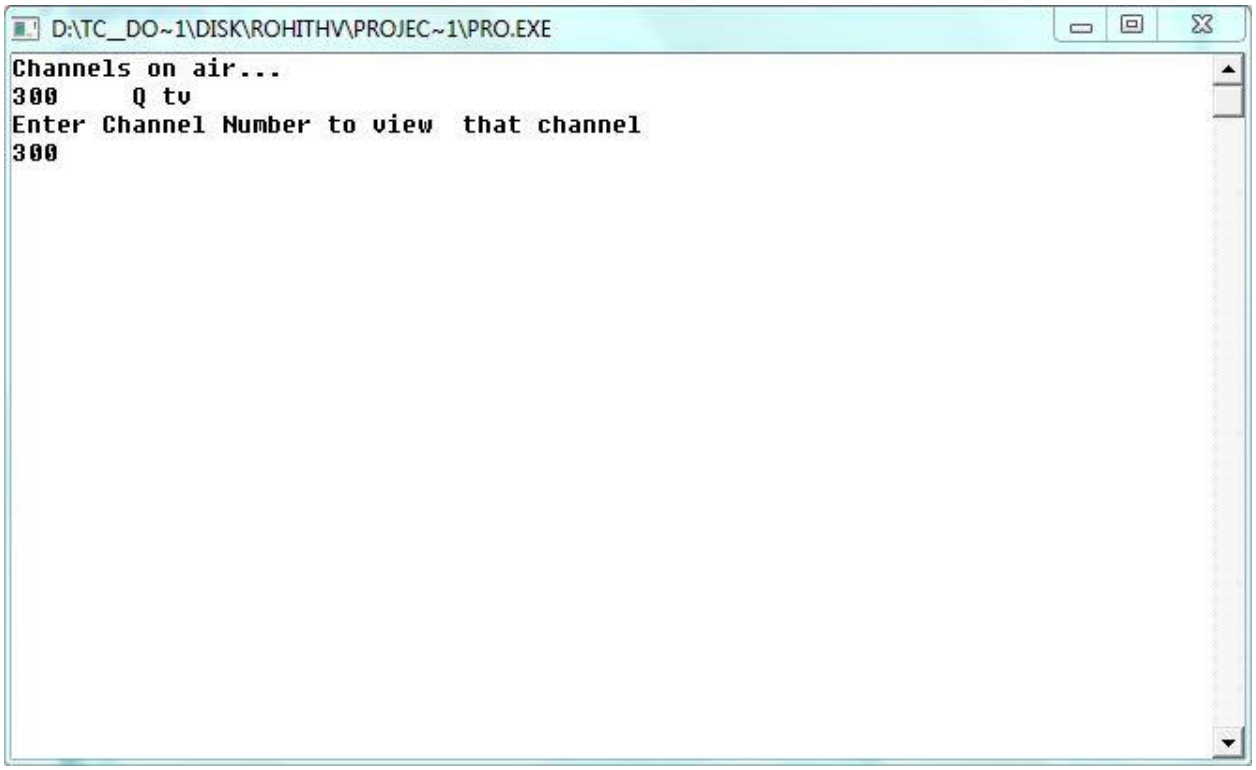
User Login:



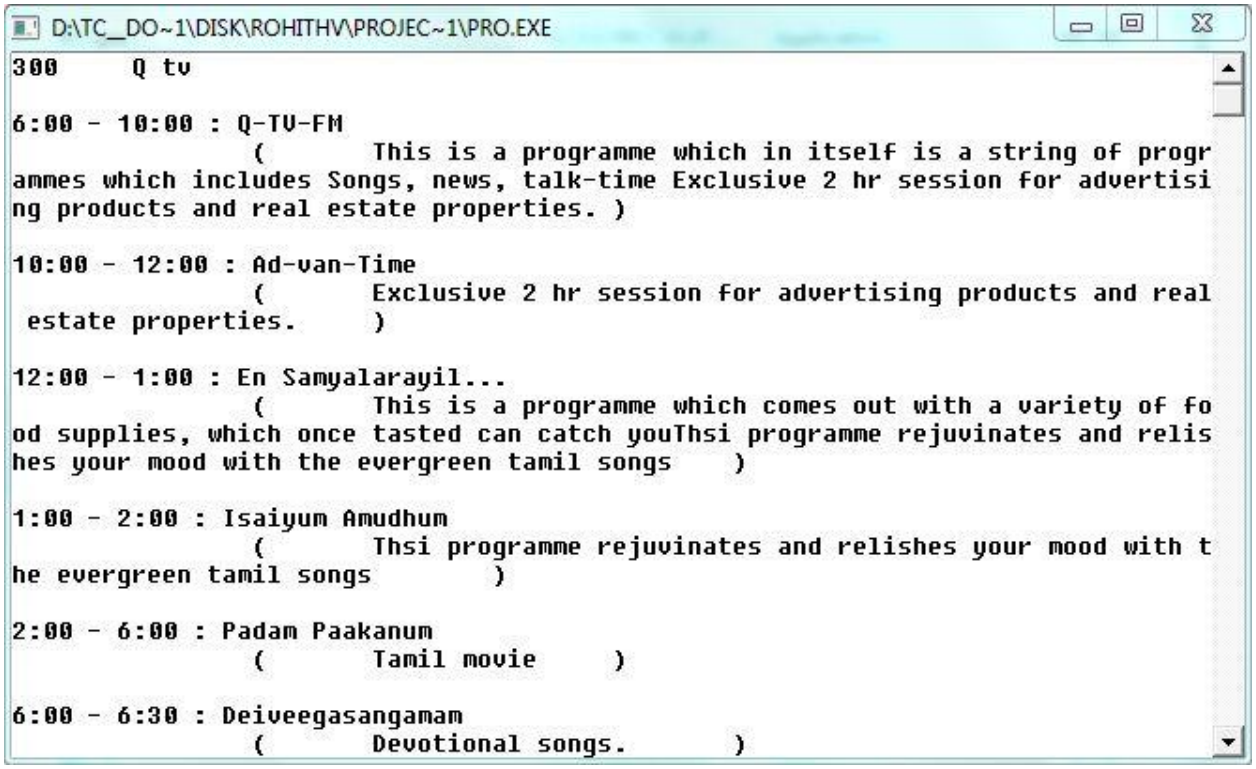
User Home Screen:



Channels on air:



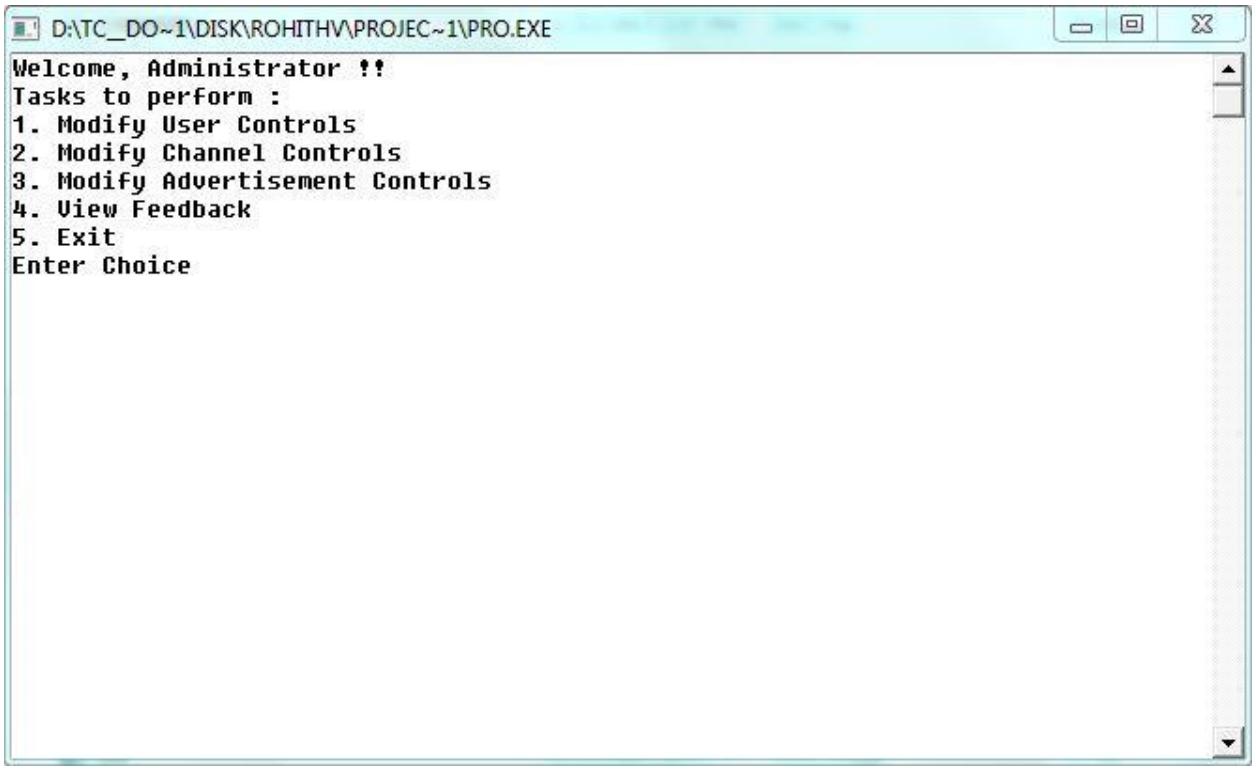
Channel View :



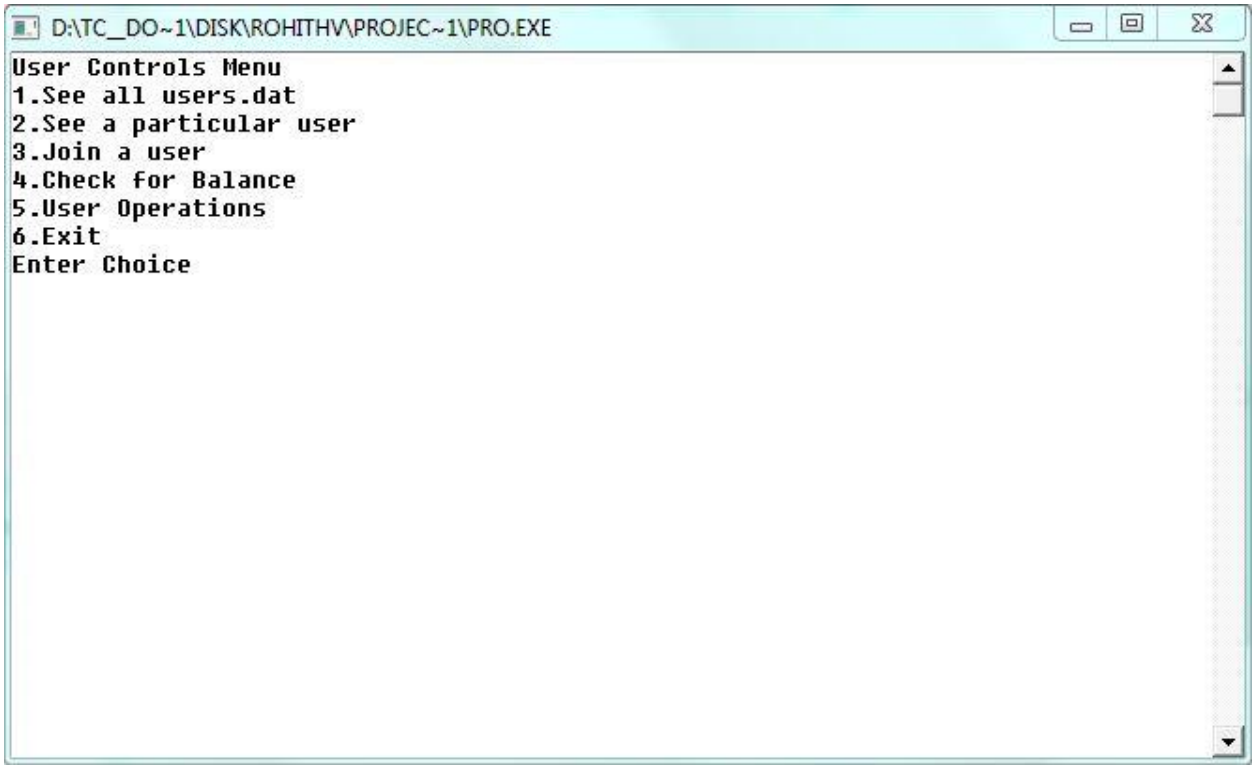
Bouquets or Brickbats:

Administrator Code:

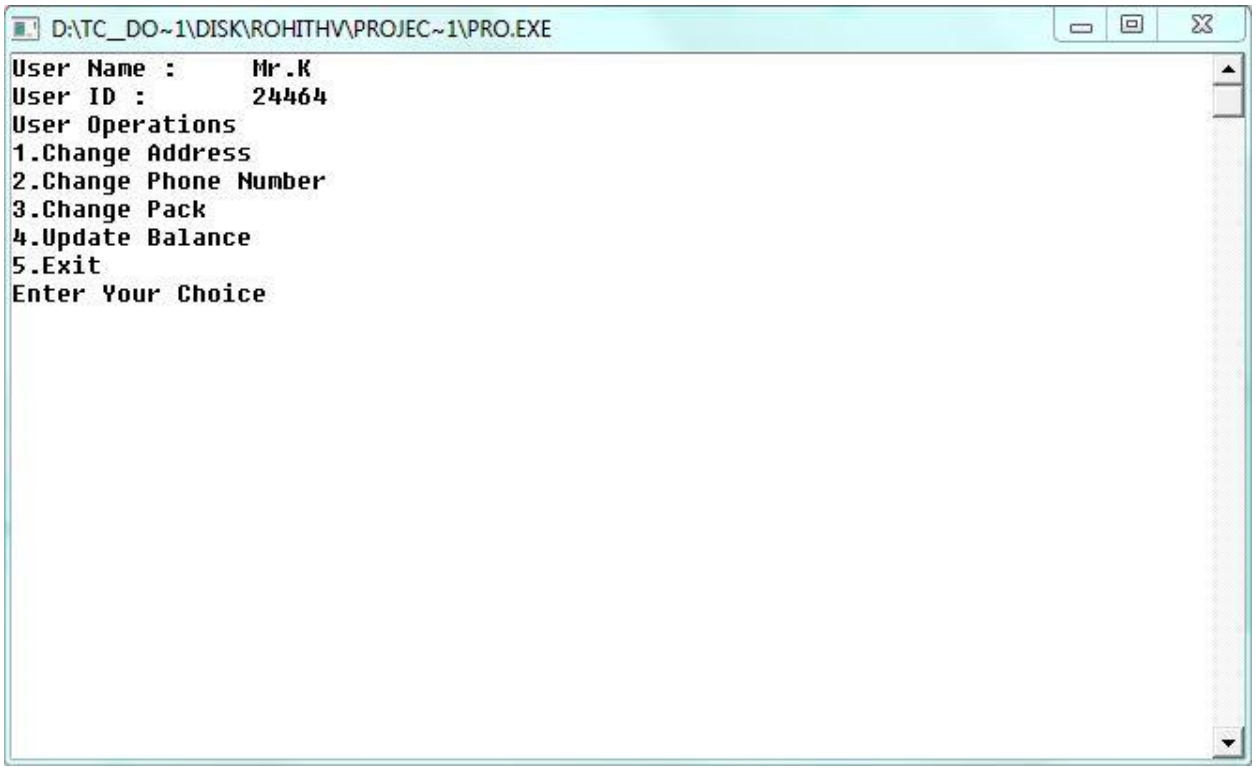
Administrator Menu:



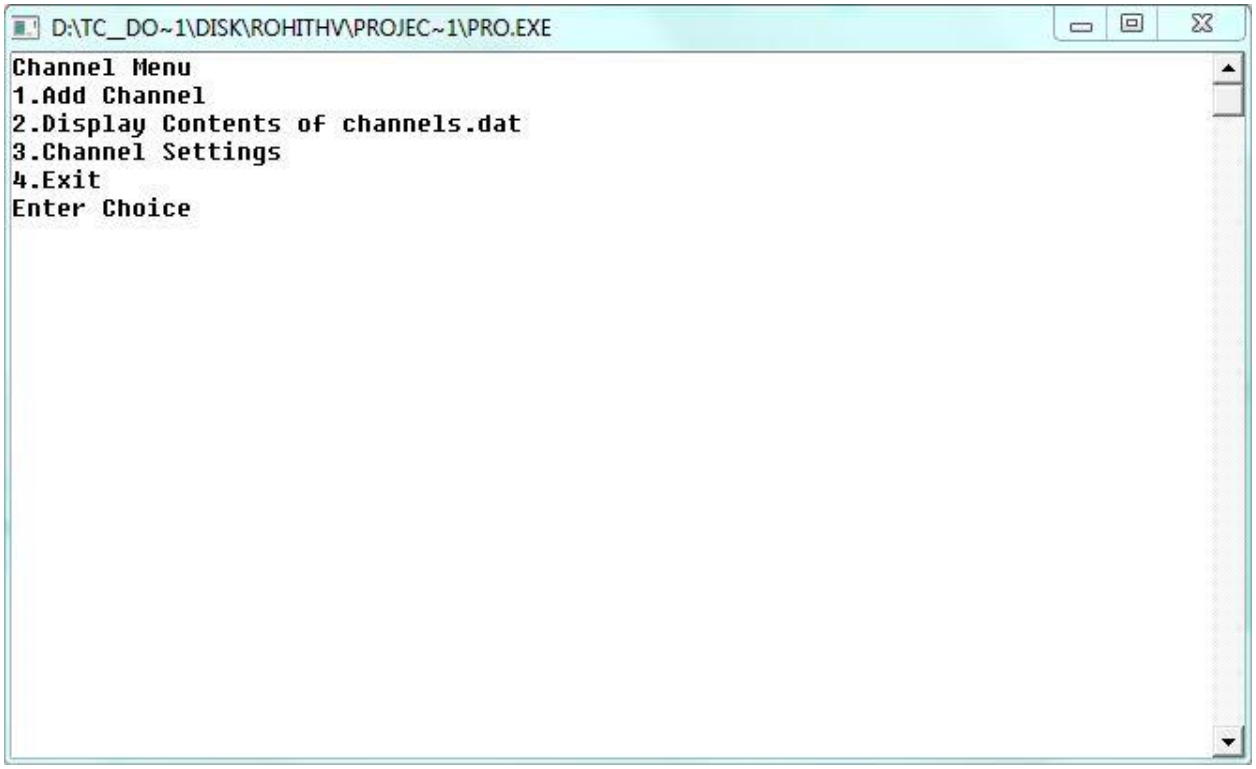
User Data Control Centre:



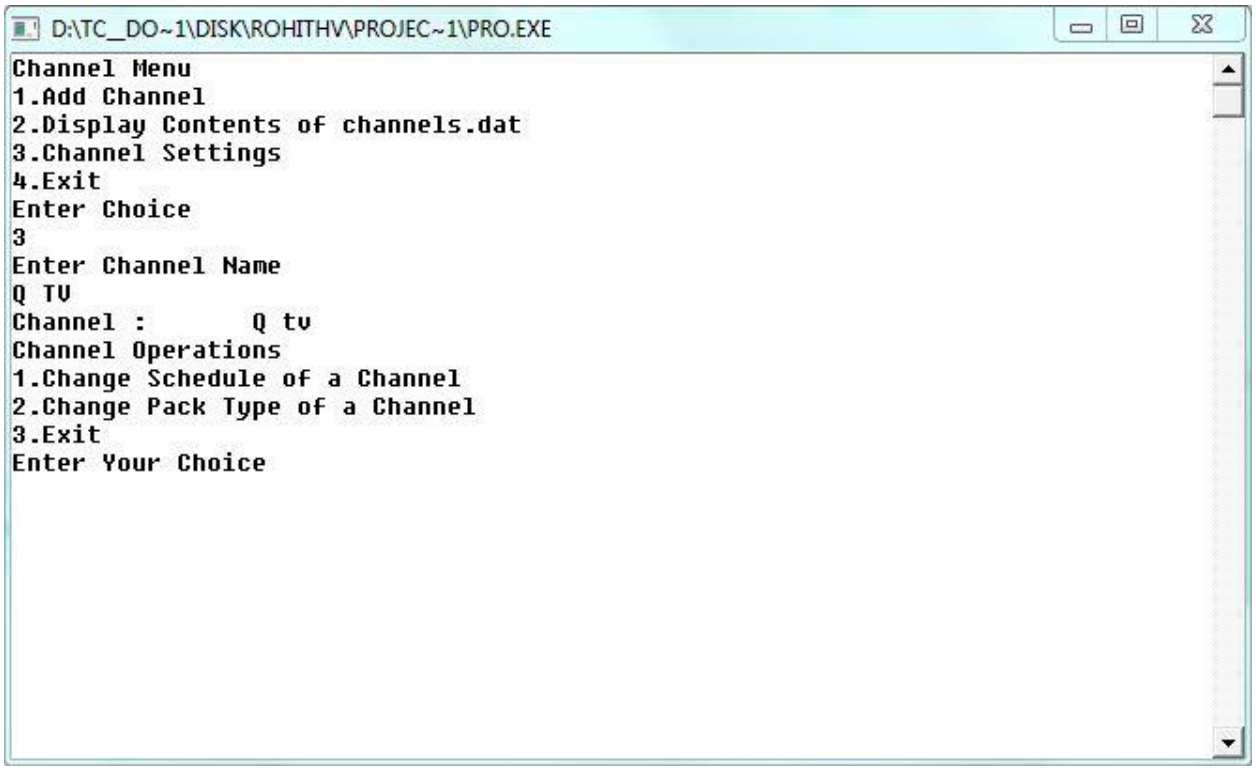
User Data Manipulation Centre:



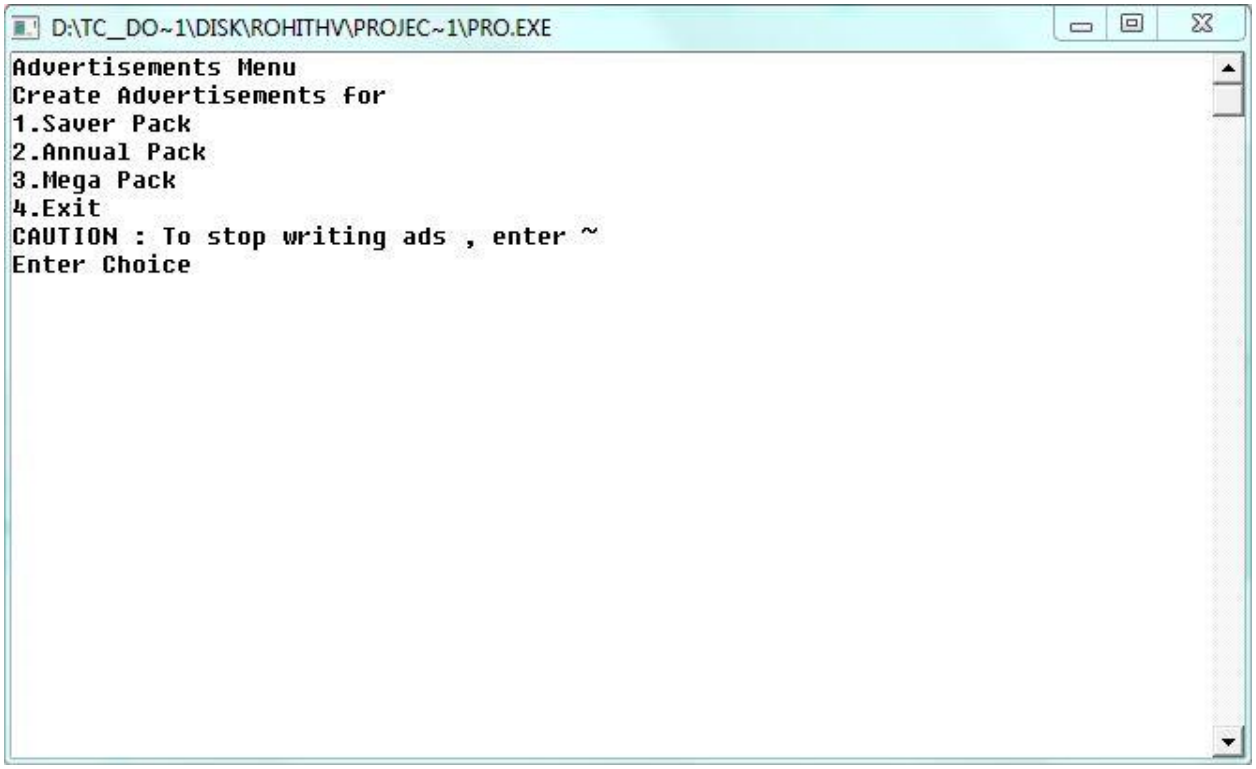
Channel Data Controls Menu:



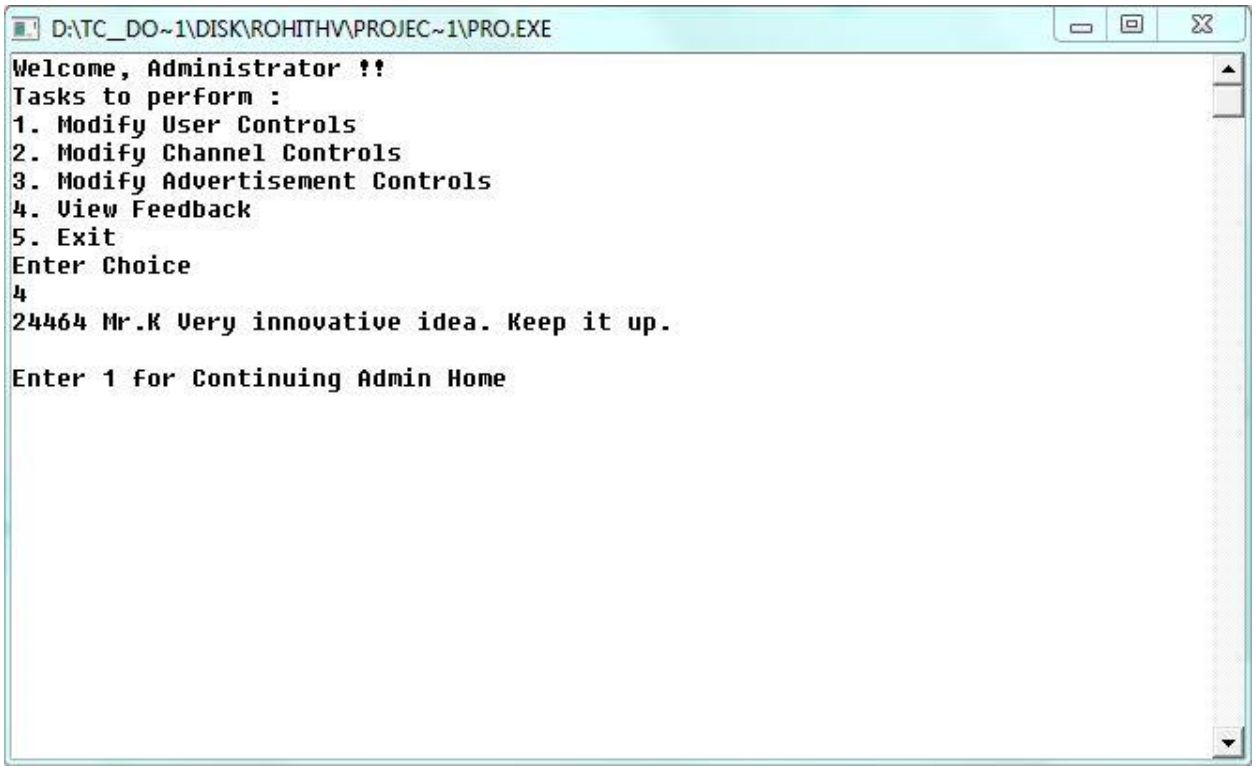
Channel Data Update Centre:



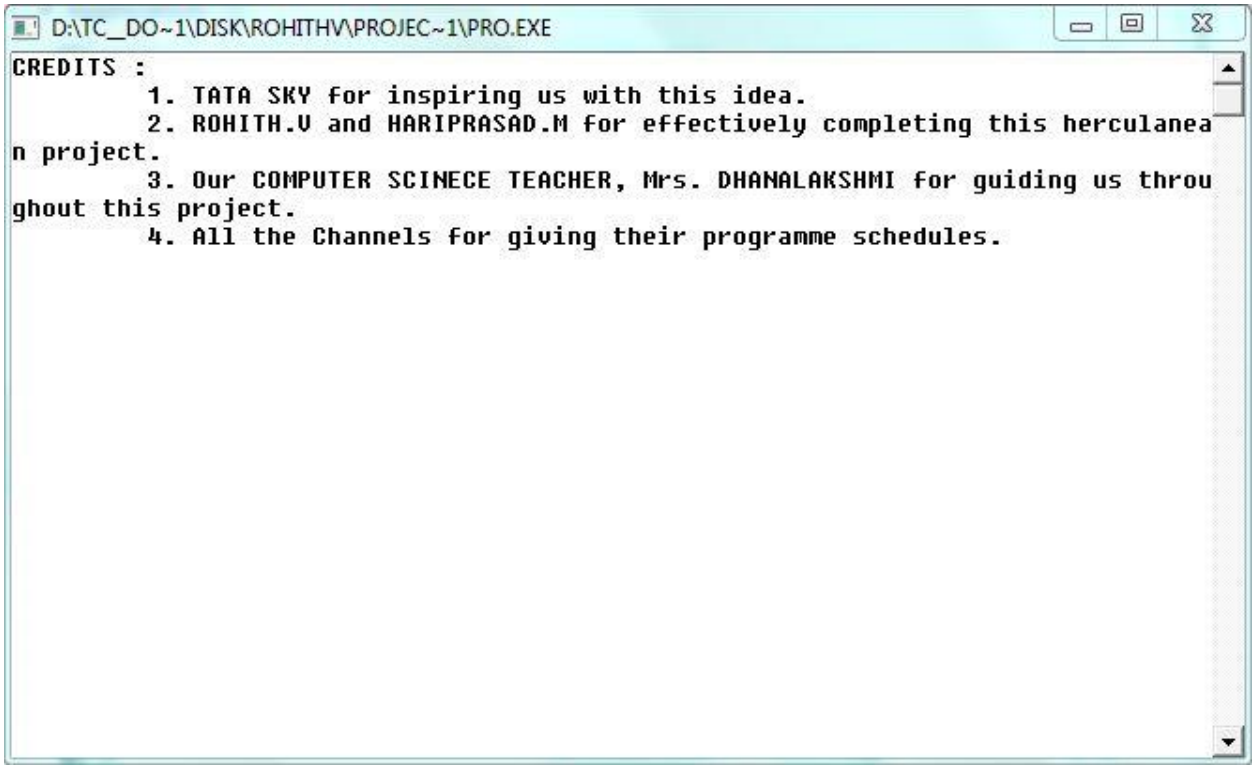
Advertisements Menu:



View the Comments:



Credits:


D:\TC__DO~1\DISK\ROHITHV\PROJEC~1\PRO.EXE
CREDITS :
1. TATA SKY for inspiring us with this idea.
2. ROHITH.V and HARIPRASAD.M for effectively completing this herculanea
n project.
3. Our COMPUTER SCINECE TEACHER, Mrs. DHANALAKSHMI for guiding us throu
ghout this project.
4. All the Channels for giving their programme schedules.

CONCLUSION

This project can be updated by implementing in the following ideas:

- Day wise schedule can be incorporated into the schedule.
- Games and other services can be added for the user at an additional cost.
- Graphics can be done.

BIBLIOGRAPHY

- Computer Science with C++ by SUMITA ARORA.
- Let Us C++ by YESHWANT KANETKAR.