

INTRODUCTION

Absolute Portfolio Manager is a stock management software with a DOS interface. It can efficiently manage many portfolios and can execute various commands that manipulate it.

APM keeps one updated with the latest market values. It also auto backups the portfolio every time it exits. The interface is extremely fluid and can be controlled with the num pad. The interface is also standard i.e. the key for exit is 0 in every menu.

The program displays its output in defined forms such as tables and graphs.

The program was constructed and built in Borland C++ v5.02 using standard C++ language.



PROGRAM ENCODING

The program is made from 5 c++ code files consisting of two source files and 3 header files:-

1. main.cpp
2. transaction.h
3. portfolio.h
4. interface.h
5. piechart.cpp

1. MAIN.CPP

Has a simple single instruction that runs the entire program:-

Run_program(): Defined in interface.h

main.cpp includes interface.h and portfolio.h

2. TRANSACTION.H

It contains various global functions and global variable declaration along with the class declarations of Transaction, holding and dividend.

Global Variables:

- Transaction Consatants(BUY, SELL, BONUS, SPLIT, NSE, BSE)
- Table Border character constants(t_l, t_r, b_l, b_r, h, v, t_s, b_s, v_t, h_t, r_s, l_s, fs_t)
- Interface variables(title, port_add, div_add, hold_add, sett_add, STT, SER, cash, brokerate)

Global Functions:

- **void Create_defaultSettings(char *settadd):** This creates defaults for settings when a new portfolio is created.
- **int sizeoffile(char add[]):** Checks whether the given file exists or not and returns the file size.
- **bool Date_formatcheck(char date[]):** Checks whether the given date is of right format and returns a true if date is right else false.
- **int date_compare(char d1[], char d2[]):** Compares the two dates and returns 1 if d1 is sooner the d2, -1 if d1 is later than d2 else returns 0 if they are the same.
- **void Render_Title(char title[], int col=YELLOW):** Displays the title on the screen with the specified color.
- **void Enter_command(int &op):** This function ask the user to enter an integer.
- **void Create_batchfile_NSE():** Creates the dos batch file containing a set of instructions for updating latest NSE prices for dos to execute.

- `void Create_batfile_BSE():()`: Creates the dos batch file containing a set of instructions for updating latest BSE prices for dos to execute.
- `int Download_Bhavcopy()`: Executes the two batch files created by the two above functions and creates the files bhavcopy_BSE.csv and bhavcopy_NSE.csv.

class Transaction:

- private members: exchange, type, name, date, quantity, rate, amount, total ratio, multiplier.
- public functions:
 - `Transaction()`: Assigns default values to the transaction object
 - `void Input()` : Inputs values into the transaction object based on its type and exchange and calculates accordingly.
 - `int get_type()`: Returns the type.
 - `int get_exchange()`: Returns the exchange.
 - `char *get_date()`: Returns the date.
 - `int get_quantity()`: Returns the quantity
 - `float get_rate()`: Returns the rate.
 - `float multi()`: Returns the multiplier constant.
 - `float get_total()`: Returns the total.
 - `char *get_name()`: Returns the name of the scrip.

Class Holding:

- private members: name
- public members:
 - `Holding()`: Initiates the holding object.
 - `Holding(char c[])`: Initiates the holding object with the provided scrip name.
 - `char* Get_name()`: Returns the name.

Class Dividend:

- private members: name, date, amount.
- public members:
 - `Dividend()`: Initiates the dividend object.
 - `void Input()`: Inputs the values into the object.
 - `char* Get_name()`: Returns the name.
 - `char* Get_date()`: Returns the date.
 - `float Get_amount()`: Returns the amount.

3. PORTFOLIO.H

It contains the portfolio class which is the core of all commands and database handling.

class Portfolio:

- private members: buyrate, sellrate, change, netquantity, buyquantity, value, gain, gainperc, curval, , port_file, div_file, hold_file, sett_file.
- Public members
- **Portfolio()**: Initiates the portfolio object to default values.
- **Portfolio(char *portfolio, char *holdings, char *dividend, char *settings)**: Initiates the portfolio object to the given parameters.
- **void Input(Transaction t)**: Inputs a transaction into the port_file according to its date. The holdings are updated and the cash balance is modified if necessary.
- **void Input(Dividend d)**: Inputs a dividend into the div_file according to its date.
- **int Output_transaction_history(char scripname[])**: Displays the transaction history of that particular scrip provided as parameter.
- **float Get_cmp_n(char scripname[], int val)**: The current market price is fetched from the bhavcopy files placed in the directory for that NSE scrip.
- **float Get_cmp_b(char Scripname[], int val)**: The current market price is fetched from the bhavcopy files placed in the directory for that BSE scrip.
- **int Output_holding(float cmp, float cmp_b, Holding h, float &inv, float &today_gain, float &today_value)**: The portfolio output for the particular scrip is displayed onto the screen. This function helps in displaying the Portfolio characteristics on the screen too.
- **void Output_portfolio()**: This outputs the entire portfolio in table form for each holding and also provides comparison with the latest values.
- **float Output_holding_pie(float cmp, float cmp_b, Holding h, float &port_perc)**: Assesses the total portfolio value for the scrip and returns this value.
- **void Output_portfolio_pie()**: Write values returned by the above function onto a text file which is to be read by piechart.exe for constructing a pie graph. The name and portfolio value are written onto the file.
- **int Output_holding_pie2(Holding h)**: This functions the net quantity for the particular scrip.
- **void Update_holdings(char c[])**: This functions updates the holding file when any new transaction is added if necessary.
- **int Delete_transaction(char *scripname, int sl_no=-1)**: Deletes a transaction based on the scrip name and the serial number provided. If sl_no=-1 then all transactions of the particular scrip are deleted.
- **void Delete_holding(char *scripname)**: Deletes a holding based on the scrip name

- `void Replace_transaction(Transaction tmp, char *scripname, int sl_no):` Replaces the transaction with the one provided.
- `void Replace_dividend(Dividend tmp, char *scripname, int sl_no):` Replaces the dividend with the one provided.
- `int Delete_dividend(char *scripname, int sl_no=-1):` Deletes a dividend based on the scrip name and the serial number provided. If `sl_no=-1` then all dividend of the particular scrip are deleted.
- `float Get_dividend(char *scripname):` Returns the total value of dividend for the scrip.
- `float Get_dividend():` Returns the total dividend of all holdings that the user has.
- `int Output_dividend_history(char scripname[]):` Outputs the dividend history for the scrip in table form and date wise.
- `void Output_dividend_date(char d1[],char d2[]):` Outputs dividends that were created within the dates `d1` and `d2` in table form.
- `void Output_dividend_all():` Outputs all dividends in table form.
- `void Load_Settings():` Load settings from file into the variables.
- `void Rewrite_Settings():` Rewrite the settings from the variables into the file.
- `int Check_holding(char c[]):` Confirms whether the given scrip exists or not.

4. INTERFACE.H

Contains the definitions of all functions that run the interface and includes the `Run_program()` function. This is the core of the interface.

Interface functions:

- `void Run_program():` This the main loop for the entire interfaces and menus.
- `void Create_Portfolio():` This functions contains the new portfolio creation interface.
- `void View_Portfolio(char *title):` This function contains the portfolio menu which includes the Transaction history menu and the Settings menu interface.
- `void Delete_Portfolio():` This functions includes the deletion of portfolio interface.
- `void Run_dividend(Portfolio &p):` This includes the dividend menu and the dividend history menu interface.
- `void Backup_Portfolio(char *title, int flg=0):` This function backups a portfolio.
- `void Restore_Portfolio(char *title):` Restores the portfolio from backup.

5. PIECHART.CPP

This is the source for the piechart.exe which draws the pie graph based on the portfolio values. It uses the dos bgi graphics.

CURL AND UNZIP

curl.exe and **unzip.exe** are the executables placed in the same directory. Curl helps to download files off the internet. Unzip extracts files from zipped files.

Dos command to download file from site:-

curl -o file.dat http://www.site.com/file.dat

But if the file is to be downloaded with a different name:-

curl -o name.dat http://www.site.com/file.dat

Dos command to unzip a zip file file.zip:-

unzip file.zip

These programs have been used to gather the latest market values from the NSE and BSE sites.

SOURCE CODE

1. MAIN.CPP

```
#include <iostream.h>
#include <fstream.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <iomanip.h>
#include <stdio.h>
#include <math.h>
#include <dos.h>
#include <ctype.h>

#include "interface.h"
#include "portfolio.h"

int main()
{
    Run_program();
    return 0;
}
```

2. TRANSACTION.H

```
#ifndef TRANSACTION
#define TRANSACTION
#include <stdio.h>
#include "portfolio.h"

#define BUY          1
#define SELL         2
#define BONUS        3
#define SPLIT        4
#define DIVID        5

#define NSE          1
#define BSE          2

/
/~~~~~GLOBALS~~~~~
char title[64];
char port_add[64];
char hold_add[64];
char divi_add[64];
char sett_add[64];
float STT;           // securities transaction tax (0.125% of buy or
                    // sell value, incl brokerage)
float SER;           // service tax (12.25% of brokerage)
float cash;
float brokrate;
```

```

char t_l=char(201);
char t_r=char(187);
char b_l=char(200);
char b_r=char(188);
char h=char(205);
char v=char(186);
char t_s=char(209);
char b_s=char(207);
char v_t=char(179);
char h_t=char(196);
char l_s=char(199);
char r_s=char(182);
char fs_t=char(197);

//~~~~~FUNCTIONS~~~~~

void Create_defaultSettings(char *settadd)
{
    fstream file(settadd, ios::out|ios::trunc);
    file<<"0.125"<<endl<<"0.1225"<<endl<<"0"<<endl<<"0.75"<<endl;
    file.close();
}

int sizeoffile(char add[])
{
    char c='a';int count=0;
    fstream file(add, ios::in|ios::nocreate);
    if(!file)return 0;
    else
    {
        while(!file.eof())
        {
            file.get(c);
            count++;
        };
        file.close();
        return count;
    }
}

bool Date_formatcheck(char date[])
{
    bool flag=true;
    if(strlen(date)!=8)flag=false;
    else if(isdigit(date[0])==0 && isdigit(date[1])==0)flag=false;
    else if(isdigit(date[3])==0 && isdigit(date[4])==0)flag=false;
    else if(isdigit(date[6])==0 && isdigit(date[7])==0)flag=false;
    else if(date[2]!='-' && date[5]!='-')flag=false;

    return flag;
}

int date_compare(char d1[], char d2[])
{
    char a[2];

```



```

int x[3],y[3];

for(int i=0;i<3;i++)
{
    a[0]=d1[i*3];
    a[1]=d1[(i*3)+1];
    x[i]=atoi(a);
    a[0]=d2[i*3];
    a[1]=d2[(i*3)+1];
    y[i]=atoi(a);
}
if(y[2]>x[2]) return 1;
else if(y[2]==x[2])
{
    if(y[1]>x[1]) return 1;
    else if(y[1]==x[1])
    {
        if(y[0]>x[0]) return 1;
        else if(y[0]==x[0]) return 0;
        else return -1;
    }
    else return -1;
}
else return -1;
}

void Render_Title(char title[], int col=YELLOW)
{
    textcolor(col);
    gotoxy(39-(strlen(title)/2),
wherey());cprintf("%s",title);cout<<endl;
    int n=strlen(title);
    if(n%2!=0)
    {
        gotoxy((38-(n/2)), wherey());
        for(int i=0;i<(n+2);i++){cprintf("~");}
    }
    else
    {
        gotoxy((38-(n/2)), wherey());
        for(int i=0;i<(n+2);i++){cprintf("~");}
    }
    cout<<endl;
}

void Enter_command(int &op)
{
    cout<<"\n\n";
    textcolor(GREEN);
    cprintf("Enter Command: [ ]");
    gotoxy(wherex()-2,wherey());
    cin>>op;
}

void Create_batfile_NSE()
{

```

```

dosdate_t temp;
_dos_getdate(&temp);
int year=temp.year;
int month=(int)temp.month;
int day=((int)temp.day);
int day_week=((int)temp.dayofweek);
char *mon;

if(day_week==0) day-=2;
else if(day_week==6) day-=1;

ofstream file("tempEX.bat",ios::out|ios::trunc);
file<<"curl -o bhavcopy_NSEtemp.csv
http://www.nseindia.com/content/historical/EQUITIES/";
file<<year<<"/";

if(month==1) mon="JAN";
else if(month==2) mon="FEB";
else if(month==3) mon="MAR";
else if(month==4) mon="APR";
else if(month==5) mon="MAY";
else if(month==6) mon="JUN";
else if(month==7) mon="JUL";
else if(month==8) mon="AUG";
else if(month==9) mon="SEP";
else if(month==10) mon="OCT";
else if(month==11) mon="NOV";
else if(month==12) mon="DEC";

file<<mon<<"/cm";
if(day>10) file<<day;
else file<<"0"<<day;
file<<mon<<year<<"bhav.csv\n";
file.close();
}

void Create_batfile_BSE()
{
dosdate_t temp;
_dos_getdate(&temp);
int year=temp.year;
year-=2000;
int month=(int)temp.month;
int day=(int)temp.day;
int day_week=((int)temp.dayofweek);
if(day_week==0) day-=2;
else if(day_week==6) day-=1;

ofstream file("tempEX.bat",ios::out|ios::trunc);
file<<"curl -o bhav_bse.zip http://www.bseindia.com/bhavcopy/eq";
if(day<10) file<<"0"<<day;
else file<<day;
if(month<10) file<<"0"<<month;
else file<<month;
if(year<10) file<<"0"<<year;
else file<<year;

```

```

file<<"_csv.zip\n";

    file<<"unzip bhav_bse.zip\n";
file<<"del bhav_bse.zip\n";
file<<"ren EQ";
if(day<10)file<<"0"<<day;
else file<<day;
if(month<10)file<<"0"<<month;
else file<<month;
if(year<10) file<<"0"<<year;
else file<<year;
file<<".CSV bhavcopy_BSE_tmp.csv\n";
file.close();
}

int Download_Bhavcopy()
{
    int flag=0;
    Create_batfile_NSE();
    system("tempEX.bat");
    remove("tempEX.bat");

if(sizeof(file("bhavcopy_NSEtemp.csv"))<500){remove("bhavcopy_NSEtemp.csv");flag++;}
    else if(sizeof(file("bhavcopy_NSEtemp.csv"))>=500)
    {
        remove("bhavcopy_NSE.csv");
        rename("bhavcopy_NSEtemp.csv", "bhavcopy_NSE.csv");
    }

    Create_batfile_BSE();
    system("tempEX.bat");
    remove("tempEX.bat");
if(sizeof(file("bhavcopy_BSE_tmp.csv"))>=500)
    {
        remove("bhavcopy_BSE.csv");
        rename("bhavcopy_BSE_tmp.csv", "bhavcopy_BSE.csv");
    }
    else
    {
        remove("bhavcopy_BSE_tmp.csv");
        flag++;
    }

    return flag;
}

//~~~~~

class Transaction
{
    int exchange;
    int type;
    char name[15];

```

```

char date[9];
int quantity;
float rate;
float amount;
float total;
char ratio[5];
float multiplier;

public:

    Transaction()
    {
        total=0;
        amount=0;
        quantity=0;
        rate=0;
        brokrate=0;
        multiplier=0;
        exchange=NSE;
    }

    void Input()
    {
        cout<<"\n\nEnter type of transaction:\t\t";
        cin>>type;
        if(type==1)
        {
            cout<<"Enter stock exchange:\t\t\t";
            cin>>exchange;
            cout<<"Enter scrip name:\t\t\t";
            gets(name);
            strupr(name);
            cout<<"Enter quantity purchased:\t\t";
            cin>>quantity;
            cout<<"Enter date of purchase(dd-mm-yy):\t";
            gets(date);
            cout<<"Enter purchase rate:\t\t\t";
            cin>>rate;
        }
        else if(type==2)
        {
            cout<<"Enter stock exchange:\t\t\t";
            cin>>exchange;
            cout<<"Enter scrip name:\t\t\t";
            gets(name);
            strupr(name);
            cout<<"Enter quantity sold:\t\t\t";
            cin>>quantity;
            cout<<"Enter date of sale(dd-mm-yy):\t\t";
            gets(date);
            cout<<"Enter sale rate:\t\t\t";
            cin>>rate;
        }
        else if(type==3)
        {

```

```

        cout<<"Enter scrip name:\t\t\t";
        gets(name);
        strupr(name);
        cout<<"Enter bonus ratio(x:y):\t\t\t";
        gets(ratio);
        cout<<"Enter date of bonus(dd-mm-yy):\t\t";
        gets(date);
        rate=0;
        char ch1[3];
        char ch2[3];
        int i,j;
        char q='a';
        for(i=0;q!=':';i++)
        {
            q=ratio[i];
            ch1[i]=ratio[i];
        }
        ch1[i-1]='\0';
        for(j=0,q='a';q!='\0';i++,j++)
        {
            q=ratio[i];
            ch2[j]=ratio[i];
        }
        ch2[j-1]='\0';
        float x=atoi(ch1);
        float y=atoi(ch2);
        multiplier=x/y;
    }
    else if(type==4)
    {
        cout<<"Enter scrip name:\t\t\t";
        gets(name);
        strupr(name);
        cout<<"Enter split ratio(x:y):\t\t\t";
        gets(ratio);
        cout<<"Enter date of split(dd-mm-yy):\t\t";
        gets(date);
        rate=0;
        char ch1[3];
        char ch2[3];
        int i,j;
        char q='a';
        for(i=0;q!=':';i++)
        {
            q=ratio[i];
            ch1[i]=ratio[i];
        }
        ch1[i-1]='\0';
        for(j=0,q='a';q!='\0';i++,j++)
        {
            q=ratio[i];
            ch2[j]=ratio[i];
        }
        ch2[j-1]='\0';
        float x=atoi(ch1);
        float y=atoi(ch2);
    }
}

```

```

        multiplier=y/x;
    }
    if (type==1)
    {
        float SER_value,STT_value,brok_value;
        brok_value=rate*quantity*brokrate*0.01;
        SER_value=rate*quantity*brokrate*0.01*SER;
        //STT_value=rate*quantity*(1+(brokrate/100))*STT;
        STT_value=rate*quantity*STT*0.01;

//total=(rate*quantity*(1+(brokrate/100)))+SER_value+STT_value;
        total=(rate*quantity) + brok_value + SER_value +
STT_value;
    }
    else if (type==2)
    {
        float SER_value,STT_value,brok_value;
        brok_value=rate*quantity*brokrate*0.01;
        SER_value=rate*quantity*brokrate*0.01*SER;
        //STT_value=rate*quantity*(1-(brokrate/100))*STT;
        STT_value=rate*quantity*STT*0.01;
        //total=(rate*quantity*(1-(brokrate/100)))-
SER_value-STT_value;
        total=(rate*quantity) - brok_value - SER_value -
STT_value;
    }
}

int get_type()
{
    return type;
}

int get_exchange()
{
    return exchange;
}

char *get_date()
{
    return date;
}

int get_quantity()
{
    return quantity;
}

float get_rate()
{
    return rate;
}

float multi()
{
    return multiplier;
}

float get_total()
{
    return total;
}

char *get_name()

```

```

        {
            return name;
        }
    };

class Holding
{
    char name[15];

public:
    Holding()
    {
    }

    Holding(char c[])
    {
        strcpy(name,c);
    }

    char* Get_name()
    {
        return name;
    }
};

class Dividend
{
    char name[15];
    char date[9];
    float amount;

public:
    Dividend()
    {
    }

    void Input()
    {
        cout<<"\n\nEnter the scrip
name:\t\t";gets(name);strupr(name);
        cout<<"Enter date(dd-mm-yy):\t\t";gets(date);
        cout<<"Enter dividend amount:\t\t";cin>>amount;
    }

    char* Get_name()
    {
        return name;
    }

    char* Get_date()
    {
        return date;
    }
};

```

```

        float Get_amount()
        {
            return amount;
        }
};

#endif

```

3. PORTFOLIO.H

```

#ifndef PORTFOLIO
#define PORTFOLIO

#include "transaction.h"

class Portfolio
{
    float buyvalue;
    float sellrate;
    float change;
    int netquantity;
    int buyquantity;
    float value;
    float gain;
    float gainperc;
    float curval;
    char *port_file;
    char *hold_file;
    char *divi_file;
    char *sett_file;

public:
    Portfolio()
    {
        buyvalue=0;
        value=0;
        sellrate=0;
        netquantity=0;
        buyquantity=0;
        gain=0;
        gainperc=0;
        curval=0;
    }

    Portfolio(char *portfolio, char *holdings, char *dividend,
char *settings)
    {
        divi_file=dividend;
        port_file=portfolio;
        hold_file=holdings;
        sett_file=settings;
        buyvalue=0;
        value=0;
        sellrate=0;
    }
}

```



```

        netquantity=0;
        buyquantity=0;
        gain=0;
        gainperc=0;
        curval=0;
    }

    void Input(Transaction t)
    {
        Transaction tmp;
        int flag=1;

        if(t.get_type()==1){cash-
=t.get_total();Rewrite_Settings();}

    if(t.get_type()==2){cash+=t.get_total();Rewrite_Settings();}

        fstream file(port_file,ios::binary|ios::in);
        fstream tfile("temp.dat",ios::binary|ios::out);
        if(!file)
            tfile.write((char*)&t,sizeof(Transaction));
        else
        {
            file.read((char*)&tmp,sizeof(Transaction));
            while(!file.eof())
            {
                if(date_compare(t.get_date(), tmp.get_date())==1 &&
flag!=0)
                {
                    tfile.write((char*)&t,sizeof(Transaction));
                    flag=0;
                }
                tfile.write((char*)&tmp,sizeof(Transaction));
                file.read((char*)&tmp,sizeof(Transaction));
            };

            if(flag==1)tfile.write((char*)&t,sizeof(Transaction));
        }

        file.close();
        tfile.close();

        remove(port_file);
        rename("temp.dat",port_file);

        Update_holdings(t.get_name());
    }

    void Input(Dividend d)
    {
        Dividend tmp;
        int flag=1;
        fstream file(divi_file,ios::binary|ios::in);
        fstream tfile("temp.dat",ios::binary|ios::out);
        if(!file)
            tfile.write((char*)&d,sizeof(Dividend));
    }

```

```

else
{
    file.read((char*)&tmp,sizeof(Dividend));
    while(!file.eof())
    {
        if(date_compare(d.Get_date(), tmp.Get_date())==1
&& flag!=0)
        {
            tfile.write((char*)&d,sizeof(Dividend));
            flag=0;
        }
        tfile.write((char*)&tmp,sizeof(Dividend));
        file.read((char*)&tmp,sizeof(Dividend));
    };

    if(flag==1)tfile.write((char*)&d,sizeof(Dividend));
}

file.close();
tfile.close();

remove(divi_file);
rename("temp.dat",divi_file);
}

int Output_transaction_history(char scripname[])
{
    float cmp=Get_cmp_n(scripname, 5);
    float cmp_b=Get_cmp_b(scripname,7);
    Transaction t;
    int sln=0;
    fstream file(port_file,ios::binary|ios::in);
    file.read((char*)&t,sizeof(Transaction));
    cout<<"Transaction History <<<scripname<<<> :\n\n";
    textcolor(LIGHTBLUE);textbackground(BLUE);
    cprintf("%c",t_l);
    for(int i=1;i<8;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<11;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<13;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<11;i++)cprintf("%c",h); cprintf("%c",t_r);
    cprintf("%c",v);
    textcolor(LIGHTCYAN);
    cprintf(" SL_NO
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf(" DATE
");textcolor(LIGHTBLUE);cprintf("%c",v_t); textcolor(LIGHTCYAN);
    cprintf(" QUANTITY
");textcolor(LIGHTBLUE);cprintf("%c",v_t); textcolor(LIGHTCYAN);
    cprintf(" BUY RATE
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf(" SALE RATE
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);

```

```

        cprintf(" EOD PRICE
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
        cprintf(" GAIN
");textcolor(LIGHTBLUE);cprintf("%c",v);
        cprintf("%c",l_s);
        for(int i=1;i<8;i++)cprintf("%c",h_t);
cprintf("%c",fs_t);
        for(int i=1;i<12;i++)cprintf("%c",h_t);
cprintf("%c",fs_t);
        for(int i=1;i<11;i++)cprintf("%c",h_t);
cprintf("%c",fs_t);
        for(int i=1;i<13;i++)cprintf("%c",h_t);
cprintf("%c",fs_t);
        for(int i=1;i<12;i++)cprintf("%c",h_t);
cprintf("%c",fs_t);
        for(int i=1;i<12;i++)cprintf("%c",h_t);
cprintf("%c",fs_t);
        for(int i=1;i<11;i++)cprintf("%c",h_t);
cprintf("%c",r_s);
        while(!file.eof())
        {
            if(strcmp(t.get_name(),scripname)==0)
            {
                float gain;
                sln++;
                cprintf("%c",v);
                for(int i=1;i<8;i++)cprintf(" ");
cprintf("%c",v_t);
                for(int i=1;i<12;i++)cprintf(" ");
cprintf("%c",v_t);
                for(int i=1;i<11;i++)cprintf(" ");
cprintf("%c",v_t);
                for(int i=1;i<13;i++)cprintf(" ");
cprintf("%c",v_t);
                for(int i=1;i<12;i++)cprintf(" ");
cprintf("%c",v_t);
                for(int i=1;i<12;i++)cprintf(" ");
cprintf("%c",v_t);
                for(int i=1;i<11;i++)cprintf(" ");
cprintf("%c",v);
                gotoxy(wherex(),wherey()-1);

textcolor(YELLOW);gotoxy(4,wherey());cprintf("%d",sln);
                cprintf("(");
                if(t.get_exchange()==1) cprintf("N");
                else cprintf("B");
                textcolor(YELLOW);cprintf(")");

textcolor(YELLOW);gotoxy(11,wherey());cprintf("%s",t.get_date());

                if(t.get_type()==BUY)
                {

textcolor(YELLOW);gotoxy(25,wherey());cprintf("%d",t.get_quantity());

```

```

textcolor(WHITE);gotoxy(34,wherey());cprintf("%.2f",t.get_rate());

textcolor(WHITE);gotoxy(47,wherey());cprintf("---");
    netquantity+=t.get_quantity();
    buyquantity+=t.get_quantity();
    buyvalue+=t.get_total();
    value=buyvalue/buyquantity;
    if(t.get_exchange()==1)gain=(cmp-
t.get_rate())*t.get_quantity();
    else gain=(cmp_b-
t.get_rate())*t.get_quantity();
    }
    else if(t.get_type()==SELL)
    {

textcolor(YELLOW);gotoxy(25,wherey());cprintf("%d",t.get_quantity());
    netquantity-=t.get_quantity();

textcolor(WHITE);gotoxy(34,wherey());cprintf("%.2f",value);

textcolor(WHITE);gotoxy(47,wherey());cprintf("%.2f",t.get_rate());
    gain=(t.get_rate()-
value)*t.get_quantity();

    if(netquantity!=0)

        value=((buyvalue/buyquantity*netquantity)-(t.get_total()-
(buyvalue/buyquantity*t.get_quantity())))/netquantity;
        else value=0;
        buyquantity-=t.get_quantity();
        buyvalue=value*netquantity;
    }
    else if(t.get_type()==BONUS)
    {
        int qty=0;
        qty=t.multi()*netquantity;
        netquantity+=qty;
        buyquantity+=qty;
        value=buyvalue/buyquantity;
        if(t.get_exchange()==1)gain=(cmp-
t.get_rate())*qty;
        else gain=(cmp_b-t.get_rate())*qty;

textcolor(YELLOW);gotoxy(25,wherey());cprintf("%d",qty);

textcolor(WHITE);gotoxy(34,wherey());cprintf("%.2f",t.get_rate());

textcolor(WHITE);gotoxy(47,wherey());cprintf("---");
    }
    else if(t.get_type()==SPLIT)
    {
        int qty=0;
        qty=t.multi()*netquantity;
        if(t.get_exchange()==1) gain=(cmp-
t.get_rate())*(qty-netquantity);

```

```

        else gain=(cmp_b-t.get_rate())*(qty-
netquantity);
        netquantity=qty;
        buyquantity=qty;
        value=buyvalue/buyquantity;

textcolor(YELLOW);gotoxy(25,wherey());cprintf("%d",qty);

        textcolor(WHITE);gotoxy(34,wherey());cprintf("%.2f",t.get_rate())
;

textcolor(WHITE);gotoxy(47,wherey());cprintf("---");
        }
        textcolor(YELLOW);gotoxy(59,wherey());
        if(t.get_exchange()==1) cprintf("%.2f",cmp);
        else cprintf("%.2f",cmp_b);
        if(gain>=0)textcolor(LIGHTGREEN);
        else textcolor(LIGHTRED);
        gotoxy(71,wherey());cprintf("%.2f",gain);
        textcolor(LIGHTBLUE);
        cout<<endl;
        cprintf("%c",v);
        for(int i=1;i<8;i++)cprintf(" ");

cprintf("%c",v_t);
        for(int i=1;i<12;i++)cprintf(" ");
cprintf("%c",v_t);
        for(int i=1;i<11;i++)cprintf(" ");
cprintf("%c",v_t);
        for(int i=1;i<13;i++)cprintf(" ");
cprintf("%c",v_t);
        for(int i=1;i<12;i++)cprintf(" ");
cprintf("%c",v_t);
        for(int i=1;i<12;i++)cprintf(" ");
cprintf("%c",v_t);
        for(int i=1;i<11;i++)cprintf(" ");
cprintf("%c",v);
        }
        file.read((char*)&t,sizeof(Transaction));
    };
    cprintf("%c",b_l);
    for(int i=1;i<8;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<12;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<11;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<13;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<12;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<12;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<12;i++)cprintf("%c",h);
cprintf("%c",b_s);
    for(int i=1;i<11;i++)cprintf("%c",h);
cprintf("%c",b_r);
    file.close();
    textbackground(BLACK);

```

```

        return sln;
    }

float Get_cmp_n(char scripname[], int val)
{
    fstream file("bhavcopy_NSE.csv", ios::in);
    char q='a';
    char c[100];
    char tname[15];
    char tcmp[15];
    float cmp=0;

    int i;
    while(!file.eof())
    {
        file.getline(c,100);
        q='a';
        for(i=0;q!=',';i++)
        {
            q=c[(i+1)];
            tname[i]=c[i];
        }
        tname[i]='\0';
        if(strcmp(tname,scripname)==0)
        {
            q='a';i=0;
            for(int j=0;j<val;j++,q='a')
                for(;q!=',';i++)
                    q=c[i];
            q='a';
            for(int j=0;q!=',';i++,j++)
            {
                q=c[(i+1)];
                tcmp[j]=c[i];
            }
            cmp=atof(tcmp);
        }
    };
    file.close();
    return cmp;
}

float Get_cmp_b(char Scripname[], int val)
{
    fstream file("bhavcopy_BSE.csv", ios::in);
    int flag=0;
    char q='a';
    char c[100];
    char tname[13];
    char scripname[13];
    strcpy(scripname, Scripname);
    for(int x=strlen(scripname);x<13;x++)scripname[x]=' ';
    char tcmp[13];
    float cmp=0;
    int i,j;

    while(!file.eof())

```

```

{
    file.getline(c,100);
    q='a';
    for(i=0;q!=',';i++)
        q=c[i];
    q='a';
    j=0;
    for(;q!=',';i++,j++)
    {
        q=c[i+1];
        tname[j]=c[i];
    }
    tname[strlen(scripname)]='\0';

    flag=0;
    for(int s=0;s<12;s++)
    {
        if(tname[s]!=scripname[s]){flag=1;break;}
    }
    if(flag==0)
    {
        q='a';i=0;
        for(int j=0;j<val;j++,q='a')
            for(;q!=',';i++)
                q=c[i];
        q='a';
        for(int j=0;q!=',';i++,j++)
        {
            q=c[(i+1)];
            tcmp[j]=c[i];
        }
        cmp=atof(tcmp);
    }
};
file.close();
return cmp;
}

```

```

int Output_holding(float cmp, float cmp_b, Holding h, float
&inv, float &today_gain, float &today_value)
{
    Transaction t;
    buyvalue=0;
    value=0;
    sellrate=0;
    netquantity=0;
    buyquantity=0;
    gain=0;
    gainperc=0;
    curval=0;

    float tgain=0;
    fstream fob(port_file,ios::binary|ios::in);
    fob.read((char*)&t,sizeof(Transaction));
    while(!fob.eof())
    {

```

```

        if(strcmp(t.get_name(),h.Get_name())==0)
        {
            if(t.get_type()==BUY)
            {
                netquantity+=t.get_quantity();
                buyquantity+=t.get_quantity();
                buyvalue+=t.get_total();
                value=buyvalue/buyquantity;
                if(t.get_exchange()==1)  tgain=(cmp-
t.get_rate())*t.get_quantity();
                else tgain=(cmp_b-
t.get_rate())*t.get_quantity();
                if(t.get_exchange()==1)
                {

                    today_gain+=t.get_quantity()*(Get_cmp_n(h.Get_name(),5)-
Get_cmp_n(h.Get_name(),7));

                    today_value+=t.get_quantity()*Get_cmp_n(h.Get_name(),5);
                }
                else
                {

                    today_gain+=t.get_quantity()*(Get_cmp_b(h.Get_name(),7)-
Get_cmp_b(h.Get_name(),9));

                    today_value+=t.get_quantity()*Get_cmp_b(h.Get_name(),7);
                }
            }
            else if(t.get_type()==SELL)
            {
                netquantity-=t.get_quantity();
                tgain=(t.get_rate()-
value)*t.get_quantity();

                if(netquantity!=0)

                value=((buyvalue/buyquantity*netquantity)-(t.get_total()-
(buyvalue/buyquantity*t.get_quantity())))/netquantity;
                else value=0;
                buyquantity-=t.get_quantity();
                buyvalue=value*netquantity;
                if(t.get_exchange()==1)
                {
                    today_gain-
=t.get_quantity()*(Get_cmp_n(h.Get_name(),5)-
Get_cmp_n(h.Get_name(),7));

                    today_value-
=t.get_quantity()*Get_cmp_n(h.Get_name(),5);
                }
                else
                {
                    today_gain-
=t.get_quantity()*(Get_cmp_b(h.Get_name(),7)-
Get_cmp_b(h.Get_name(),9));

                    today_value-
=t.get_quantity()*Get_cmp_b(h.Get_name(),7);

```



```

    }
}
else if (t.get_type() == BONUS)
{
    int qty=0;
    qty=t.multi()*netquantity;
    netquantity+=qty;
    buyquantity+=qty;
    value=buyvalue/buyquantity;
    if (t.get_exchange() == 1) tgain=(cmp-
t.get_rate())*qty;

    else tgain=(cmp_b-t.get_rate())*qty;
    if (t.get_exchange() == 1)
    {

today_gain+=t.get_quantity()*(Get_cmp_n(h.Get_name(),5)-
Get_cmp_n(h.Get_name(),7));

today_value+=t.get_quantity()*Get_cmp_n(h.Get_name(),5);
    }
    else
    {
        today_gain-
=t.get_quantity()*(Get_cmp_b(h.Get_name(),7)-
Get_cmp_b(h.Get_name(),9));

today_value+=t.get_quantity()*Get_cmp_b(h.Get_name(),7);
    }
}
else if (t.get_type() == SPLIT)
{
    int qty=0;
    qty=t.multi()*netquantity;
    if (t.get_exchange() == 1) tgain=(cmp-
t.get_rate())*(qty-netquantity);

    else tgain=(cmp_b-t.get_rate())*(qty-
netquantity);

    if (t.get_exchange() == 1)
    {
        today_gain+=(qty-
netquantity)*(Get_cmp_n(h.Get_name(),5)- Get_cmp_n(h.Get_name(),7));
        today_value+=(qty-
netquantity)*Get_cmp_n(h.Get_name(),5);
    }
    else
    {
        today_gain+=(qty-
netquantity)*(Get_cmp_b(h.Get_name(),7)- Get_cmp_b(h.Get_name(),9));
        today_value+=(qty-
netquantity)*Get_cmp_b(h.Get_name(),7);
    }
    netquantity=qty;
    buyquantity=qty;
    value=buyvalue/buyquantity;
}
}
}

```

```

        fob.read((char*)&t,sizeof(Transaction));
    };
    if(t.get_exchange()==1) {gain=(cmp-
value)*netquantity; /*l_ex=NSE;*/}
    else {gain=(cmp_b-value)*netquantity; /*l_ex=BSE;*/}
    if(netquantity!=0 &&
value!=0) gainperc=(gain/(value*netquantity))*100;
    if(netquantity==0)
        gain=tgain;
    curval=cmp*netquantity;
    inv+=netquantity*value;
    textcolor(LIGHTBLUE);
    cprintf("%c",v);
    for(int i=1;i<13;i++) cprintf(" "); cprintf("%c",v_t);
    for(int i=1;i<11;i++) cprintf(" "); cprintf("%c",v_t);
    for(int i=1;i<11;i++) cprintf(" "); cprintf("%c",v_t);
    for(int i=1;i<12;i++) cprintf(" "); cprintf("%c",v_t);
    for(int i=1;i<11;i++) cprintf(" "); cprintf("%c",v_t);
    for(int i=1;i<8;i++) cprintf(" "); cprintf("%c",v_t);
    for(int i=1;i<13;i++) cprintf(" "); cprintf("%c",v);
    gotoxy(wherex(),wherey()-1);

textcolor(YELLOW);gotoxy(3,wherey());cprintf("%s",h.Get_name());

textcolor(YELLOW);gotoxy(17,wherey());cprintf("%d",netquantity);

textcolor(WHITE);gotoxy(27,wherey());cprintf("%.2f",value);
textcolor(LIGHTGRAY);gotoxy(38,wherey());
if(t.get_exchange()==1) cprintf("%.2f",cmp);
else cprintf("%.2f",cmp_b);
if(gain<0)
{
    textcolor(LIGHTRED);
    gotoxy(49,wherey());cprintf("%.2f",gain);
    gotoxy(60,wherey());
    if(netquantity!=0) cprintf("%.1f",gainperc);
    else
    {
        gotoxy(61,wherey());
        cprintf("---");
    }
}
else
{
    textcolor(LIGHTGREEN);
    gotoxy(50,wherey());cprintf("%.2f",gain);
    gotoxy(61,wherey());
    if(netquantity!=0) cprintf("%.1f",gainperc);
    else cprintf("---");
}

textcolor(LIGHTGRAY);gotoxy(70,wherey());cprintf("%.2f",curval);
cout<<endl;
textcolor(LIGHTBLUE);
cprintf("%c",v);
for(int i=1;i<13;i++) cprintf(" "); cprintf("%c",v_t);

```

```

        for(int i=1;i<11;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<11;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<12;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<11;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<8;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<13;i++)cprintf(" "); cprintf("%c",v);
        fob.close();

        return netquantity;
    }

void Output_portfolio()
{
    Holding h;
    float today_gain=0,today_value=0, investment=0, last_ex;
    int qty;
    fstream file(hold_file,ios::binary|ios::in);
    textcolor(LIGHTBLUE);textbackground(BLUE);
    cprintf("%c",t_l);
    for(int i=1;i<13;i++)cprintf("%c",::h); cprintf("%c",t_s);
    for(int i=1;i<11;i++)cprintf("%c",::h); cprintf("%c",t_s);
    for(int i=1;i<11;i++)cprintf("%c",::h); cprintf("%c",t_s);
    for(int i=1;i<12;i++)cprintf("%c",::h); cprintf("%c",t_s);
    for(int i=1;i<11;i++)cprintf("%c",::h); cprintf("%c",t_s);
    for(int i=1;i<8;i++)cprintf("%c",::h); cprintf("%c",t_s);
    for(int i=1;i<13;i++)cprintf("%c",::h); cprintf("%c",t_r);
    cprintf("%c",v);
    textcolor(LIGHTCYAN);
    cprintf("    STOCK
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf("    QUANTITY
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf("    AVG RATE
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf("    EOD PRICE
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf("    GAIN
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf("    GAIN% ");
    textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf("LATEST
VALUE");textcolor(LIGHTBLUE);cprintf("%c",v);
    cprintf("%c",l_s);
    for(int i=1;i<13;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
    for(int i=1;i<11;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
    for(int i=1;i<11;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
    for(int i=1;i<12;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
    for(int i=1;i<11;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
    for(int i=1;i<8;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
    for(int i=1;i<13;i++)cprintf("%c",h_t); cprintf("%c",r_s);

    file.read((char*)&h,sizeof(Holding));
    while(!file.eof())
    {
        qty=Output_holding(Get_cmp_n(h.Get_name(),
5),Get_cmp_b(h.Get_name(), 7), h, investment, today_gain, today_value);

```

```

        file.read((char*)&h,sizeof(Holding));
    };
    textcolor(LIGHTBLUE);
    cprintf("%c",b_l);
    for(int i=1;i<13;i++)cprintf("%c",:h); cprintf("%c",b_s);
    for(int i=1;i<11;i++)cprintf("%c",:h); cprintf("%c",b_s);
    for(int i=1;i<11;i++)cprintf("%c",:h); cprintf("%c",b_s);
    for(int i=1;i<12;i++)cprintf("%c",:h); cprintf("%c",b_s);
    for(int i=1;i<11;i++)cprintf("%c",:h); cprintf("%c",b_s);
    for(int i=1;i<8;i++)cprintf("%c",:h); cprintf("%c",b_s);
    for(int i=1;i<13;i++)cprintf("%c",:h); cprintf("%c",b_r);
    textbackground(BLACK);

    if(today_value!=0 && investment!=0)
    {
        textcolor(WHITE);
        cout<<"\n\n\n";
        cprintf("Unrealised Gain Today    : ");
        if(today_gain>=0)textcolor(LIGHTGREEN);
        else textcolor(RED);
        cprintf("%.2f",today_gain);
        gotoxy(50,wherey());
        cprintf( "%.2f%%", ((today_gain/today_value)*100) );
        cout<<endl;

        textcolor(WHITE);
        cprintf("Portfolio Value Today    : ");
        textcolor(LIGHTGREEN);
        cprintf("%.2f",today_value);
        gotoxy(50,wherey());
        cprintf( "%.2f%%", (((today_value/investment)*100)-100)

    );

        cout<<endl;

        Load_Settings();
        textcolor(WHITE);
        cprintf("Balance Amount          : ");
        textcolor(LIGHTGREEN);
        cprintf("%.2f",cash);
    }
    file.close();
    textbackground(BLACK);
}

float Output_holding_pie(float cmp, float cmp_b, Holding h,
float &port_perc)
{
    Transaction t;
    netquantity=0;
    float scrip_perc=0;

    fstream fob(port_file,ios::binary|ios::in);
    fob.read((char*)&t,sizeof(Transaction));
    while(!fob.eof())
    {
        if(strcmp(t.get_name(),h.Get_name())==0)

```

```

        {
            if(t.get_type()==BUY)
            {
                netquantity+=t.get_quantity();

if(t.get_exchange()==NSE) scrip_perc+=t.get_quantity()*Get_cmp_n(t.get_n
ame(), 5);

                else
scrip_perc+=t.get_quantity()*Get_cmp_b(t.get_name(), 7);
            }
            else if(t.get_type()==SELL)
            {
                netquantity-=t.get_quantity();
                if(t.get_exchange()==NSE) scrip_perc-
=t.get_quantity()*Get_cmp_n(t.get_name(), 5);
                else scrip_perc-
=t.get_quantity()*Get_cmp_b(t.get_name(), 7);
            }
            else if(t.get_type()==BONUS)
            {
                int qty=0;
                qty=t.multi()*netquantity;
                netquantity+=qty;

if(t.get_exchange()==NSE) scrip_perc+=qty*Get_cmp_n(t.get_name(), 5);
                else
scrip_perc+=qty*Get_cmp_b(t.get_name(), 7);

            }
            else if(t.get_type()==SPLIT)
            {
                int qty=0;
                qty=t.multi()*netquantity;

if(t.get_exchange()==NSE) scrip_perc+=(qty-
netquantity)*Get_cmp_n(t.get_name(), 5);
                else scrip_perc+=(qty-
netquantity)*Get_cmp_b(t.get_name(), 7);
                netquantity=qty;
            }
        }
        fob.read((char*)&t,sizeof(Transaction));
    };

    port_perc+=scrip_perc;
    fob.close();
    return scrip_perc;
}

void Output_portfolio_pie()
{
    Holding h;
    int n=0;
    float port_perc=0;
    fstream file(hold_file,ios::binary|ios::in);
    ofstream fob("pietemp.txt");

```

```

file.read((char*)&h,sizeof(Holding));
while(!file.eof())
{
    if(Output_holding_pie2(h)>0)
    {
        fob<<h.Get_name()<<":";
        fob<<Output_holding_pie(Get_cmp_n(h.Get_name(),
5),Get_cmp_b(h.Get_name(), 7), h, port_perc);
        fob<<'\n';
    }
    file.read((char*)&h,sizeof(Holding));
    n++;
};

file.close();
fob.close();
}

int Output_holding_pie2(Holding h)
{
    Transaction t;
    netquantity=0;
    buyquantity=0;

    fstream fob(port_file,ios::binary|ios::in);
    fob.read((char*)&t,sizeof(Transaction));
    while(!fob.eof())
    {
        if(strcmp(t.get_name(),h.Get_name())==0)
        {
            if(t.get_type()==BUY)
            {
                netquantity+=t.get_quantity();
                buyquantity+=t.get_quantity();
            }
            else if(t.get_type()==SELL)
            {
                netquantity-=t.get_quantity();
                buyquantity-=t.get_quantity();
            }
            else if(t.get_type()==BONUS)
            {
                int qty=0;
                qty=t.multi()*netquantity;
                netquantity+=qty;
                buyquantity+=qty;
            }
            else if(t.get_type()==SPLIT)
            {
                int qty=0;
                qty=t.multi()*netquantity;
                netquantity=qty;
                buyquantity=qty;
            }
        }
        fob.read((char*)&t,sizeof(Transaction));
    }
}

```

```

        };
        fob.close();

        return netquantity;
    }

void Update_holdings(char c[])
{
    int flag=1;
    fstream file(hold_file,ios::binary|ios::in);

    if(!file)
    {
        file.close();
        Holding h(c);
        fstream fob(hold_file,ios::binary|ios::out);
        fob.write((char*)&h,sizeof(Holding));
        fob.close();
    }
    else
    {
        Holding h;
        file.read((char*)&h,sizeof(Holding));
        while(!file.eof())
        {
            if(strcmp(h.Get_name(), c)==0){flag=0;break;}
            file.read((char*)&h,sizeof(Holding));
        };
        file.close();
        if(flag!=0)
        {
            Holding q(c);
            fstream fob(hold_file,ios::binary|ios::app);
            fob.write((char*)&q,sizeof(Holding));
            fob.close();
        }
    }
}

int Delete_transaction(char *scripname, int sl_no=-1)
{
    ifstream file(port_file, ios::binary);
    ofstream tfile("temp.dat", ios::binary|ios::trunc);
    Transaction t;
    int sln=0, flag=0, f=0, count=0;
    file.read((char*)&t,sizeof(Transaction));
    while(!file.eof())
    {
        f=0;
        if(strcmp(t.get_name(),scripname)==0)count++;
        if(strcmp(t.get_name(),scripname)==0 && flag==0)
        {
            sln++;
            if(sl_no==--1){f=1;}
            else if(sln==sl_no){flag=1;f=1;}
        }
    }
}

```

```

        if(f==0)tfile.write((char*)&t,sizeof(Transaction));
        file.read((char*)&t,sizeof(Transaction));
    };

    file.close();
    tfile.close();

    remove(port_file);
    rename("temp.dat",port_file);
    if(sl_no==-1 || count==1){Delete_holding(scripname);return
0;}

    else return 1;
}

void Delete_holding(char *scripname)
{
    ifstream file(hold_file, ios::binary);
    ofstream tfile("temp.dat", ios::binary|ios::trunc);
    Holding h;
    int flag=0;
    file.read((char*)&h, sizeof(Holding));
    while(!file.eof())
    {
        flag=0;
        if(strcmp(h.Get_name(),scripname)==0){flag=1;}
        if(flag==0)tfile.write((char*)&h, sizeof(Holding));
        file.read((char*)&h, sizeof(Holding));
    };

    file.close();
    tfile.close();

    remove(hold_file);
    rename("temp.dat",hold_file);
}

void Replace_transaction(Transaction tmp, char *scripname, int
sl_no)
{
    Delete_transaction(scripname, sl_no);
    Input(tmp);
}

//~~~~~DIVIDENDS~~~~~

void Replace_dividend(Dividend tmp, char *scripname, int
sl_no)
{
    Delete_dividend(scripname, sl_no);
    Input(tmp);
}

int Delete_dividend(char *scripname, int sl_no=-1)
{
    ifstream file(divi_file, ios::binary);

```



```

ofstream tfile("temp.dat", ios::binary|ios::trunc);
Dividend d;
int sln=0, flag=0, f=0, count=0;
file.read((char*)&d,sizeof(Dividend));
while(!file.eof())
{
    f=0;
    if(strcmp(d.Get_name(),scripname)==0) count++;
    if(strcmp(d.Get_name(),scripname)==0 && flag==0)
    {
        sln++;
        if(sl_no==-1){f=1;}
        else if(sln==sl_no){flag=1;f=1;}
    }
    if(f==0)tfile.write((charsizeof(Dividend));
    file.read((charsizeof(Dividend));
};

file.close();
tfile.close();

remove(divi_file);
rename("temp.dat",divi_file);
if(sl_no==-1 || count==1){Delete_holding(scripname);return
0;}

else return 1;
}

float Get_dividend(char *scripname)
{
    float sum=0;
    Dividend d;
    ifstream file(divi_file, ios::binary);

    file.read((charsizeof(Dividend));
    while(!file.eof())
    {
        if(strcmp(scripname,d.Get_name())==0)
        {
            sum+=d.Get_amount();
        }
        file.read((charsizeof(Dividend));
    };

    file.close();
    return sum;
}

float Get_dividend()
{
    float sum=0;
    Dividend d;
    ifstream file(divi_file, ios::binary);

    file.read((charsizeof(Dividend));
    while(!file.eof())

```

```

    {
        sum+=d.Get_amount();
        file.read((char*)&d, sizeof(Dividend));
    };

    file.close();
    return sum;
}

int Output_dividend_history(char scripname[])
{
    Dividend d;
    int sln=0;
    ifstream file(divi_file, ios::binary);

    cout<<"Dividend History <"<<scripname<<"> :\n\n";

    textcolor(LIGHTBLUE);textbackground(BLUE);
    printf("%c",t_l);
    for(int i=1;i<8;i++)printf("%c",h); printf("%c",t_s);
    for(int i=1;i<12;i++)printf("%c",h); printf("%c",t_s);
    for(int i=1;i<16;i++)printf("%c",h); printf("%c",t_r);
    cout<<endl;
    printf("%c",v);
    textcolor(LIGHTCYAN);
    printf(" SL_NO
");textcolor(LIGHTBLUE);printf("%c",v_t);textcolor(LIGHTCYAN);
    printf(" DATE
");textcolor(LIGHTBLUE);printf("%c",v_t); textcolor(LIGHTCYAN);
    printf(" AMOUNT
");textcolor(LIGHTBLUE);printf("%c",v);
    cout<<endl;
    printf("%c",v);
    for(int i=1;i<8;i++)printf("%c",h_t); printf("%c",fs_t);
    for(int i=1;i<12;i++)printf("%c",h_t); printf("%c",fs_t);
    for(int i=1;i<16;i++)printf("%c",h_t); printf("%c",v);
    cout<<endl;

    file.read((char*)&d, sizeof(Dividend));
    while(!file.eof())
    {
        if(strcmp(d.Get_name(),scripname)==0)
        {
            sln++;
            textcolor(LIGHTBLUE);
            printf("%c",v);
            for(int i=1;i<8;i++)printf(" "); printf("%c",v_t);
            for(int i=1;i<12;i++)printf(" "); printf("%c",v_t);
            for(int i=1;i<16;i++)printf(" "); printf("%c",v);

textcolor(YELLOW);gotoxy(4,wherey());printf("%d",sln);

textcolor(YELLOW);gotoxy(11,wherey());printf("%s",d.Get_date());

```

```

textcolor(LIGHTGREEN);gotoxy(26,wherey());cprintf("%.2f",d.Get_amount()
);

        cout<<endl;
        textcolor(LIGHTBLUE);
        cprintf("%c",v);
        for(int i=1;i<8;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<12;i++)cprintf(" "); cprintf("%c",v_t);
        for(int i=1;i<16;i++)cprintf(" "); cprintf("%c",v);
        cout<<endl;
    }
    file.read((char*)&d, sizeof(Dividend));
};
textcolor(LIGHTBLUE);
cprintf("%c",b_l);
for(int i=1;i<8;i++)cprintf("%c",h); cprintf("%c",b_s);
for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",b_s);
for(int i=1;i<16;i++)cprintf("%c",h); cprintf("%c",b_r);
cout<<"\n\n";

file.close();
textbackground(BLACK);
textcolor(GREEN);cprintf("Total Dividend: %.2f",
Get_dividend(scripname));

    return sln;
}

void Output_dividend_date(char d1[],char d2[])
{
    int sln=0;
    float total=0;
    Dividend d;
    ifstream file(divi_file,ios::binary);

    cout<<"Dividend History:\n\n";

    textcolor(LIGHTBLUE);textbackground(BLUE);
    cprintf("%c",t_l);
    for(int i=1;i<8;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<13;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",t_s);
    for(int i=1;i<16;i++)cprintf("%c",h); cprintf("%c",t_r);
    cout<<endl;
    cprintf("%c",v);
    textcolor(LIGHTCYAN);
    cprintf(" SL_NO
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf(" STOCK
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
    cprintf(" DATE
");textcolor(LIGHTBLUE);cprintf("%c",v_t); textcolor(LIGHTCYAN);
    cprintf(" AMOUNT
");textcolor(LIGHTBLUE);cprintf("%c",v);
    cout<<endl;

```

```

        cprintf("%c",v);
        for(int i=1;i<8;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
        for(int i=1;i<13;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
        for(int i=1;i<12;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
        for(int i=1;i<16;i++)cprintf("%c",h_t); cprintf("%c",v);
        cout<<endl;

        file.read((char*)&d,sizeof(Dividend));
        while(!file.eof())
        {

if((date_compare(d1,d.Get_date())>=0)&&(date_compare(d2,d.Get_date())<=
0))
        {
                sln++;
                total+=d.Get_amount();
                textcolor(LIGHTBLUE);
                cprintf("%c",v);
                for(int i=1;i<8;i++)cprintf(" "); cprintf("%c",v_t);
                for(int i=1;i<13;i++)cprintf(" "); cprintf("%c",v_t);
                for(int i=1;i<12;i++)cprintf(" "); cprintf("%c",v_t);
                for(int i=1;i<16;i++)cprintf(" "); cprintf("%c",v);

textcolor(YELLOW);gotoxy(4,wherey());cprintf("%d",sln);

textcolor(YELLOW);gotoxy(10,wherey());cprintf("%s",d.Get_name());

textcolor(YELLOW);gotoxy(24,wherey());cprintf("%s",d.Get_date());

textcolor(LIGHTGREEN);gotoxy(41,wherey());cprintf("%.2f",d.Get_amount()
);

                cout<<endl;
                textcolor(LIGHTBLUE);
                cprintf("%c",v);
                for(int i=1;i<8;i++)cprintf(" "); cprintf("%c",v_t);
                for(int i=1;i<13;i++)cprintf(" "); cprintf("%c",v_t);
                for(int i=1;i<12;i++)cprintf(" "); cprintf("%c",v_t);
                for(int i=1;i<16;i++)cprintf(" "); cprintf("%c",v);
                cout<<endl;
        }

        file.read((char*)&d, sizeof(Dividend));
};

textcolor(LIGHTBLUE);
cprintf("%c",b_l);
for(int i=1;i<8;i++)cprintf("%c",h); cprintf("%c",b_s);
for(int i=1;i<13;i++)cprintf("%c",h); cprintf("%c",b_s);
for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",b_s);
for(int i=1;i<16;i++)cprintf("%c",h); cprintf("%c",b_r);
cout<<"\n\n";

file.close();
textbackground(BLACK);

```

```

        textcolor(GREEN);cprintf("Total Dividend: %.2f", total);

        cout<<"\n\n\n";
        if(sln==0)
            cprintf("No Dividend records during this period!");
    }

    void Output_dividend_all()
    {
        Dividend d;
        int sln=0;
        ifstream file(divi_file,ios::binary);
        cout<<"All Dividends:\n\n";

        textcolor(LIGHTBLUE);textbackground(BLUE);
        cprintf("%c",t_l);
        for(int i=1;i<8;i++)cprintf("%c",h); cprintf("%c",t_s);
        for(int i=1;i<13;i++)cprintf("%c",h); cprintf("%c",t_s);
        for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",t_s);
        for(int i=1;i<16;i++)cprintf("%c",h); cprintf("%c",t_r);
        cout<<endl;
        cprintf("%c",v);
        textcolor(LIGHTCYAN);
        cprintf(" SL_NO
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
        cprintf(" STOCK
");textcolor(LIGHTBLUE);cprintf("%c",v_t);textcolor(LIGHTCYAN);
        cprintf(" DATE
");textcolor(LIGHTBLUE);cprintf("%c",v_t); textcolor(LIGHTCYAN);
        cprintf(" AMOUNT
");textcolor(LIGHTBLUE);cprintf("%c",v);
        cout<<endl;
        cprintf("%c",v);
        for(int i=1;i<8;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
        for(int i=1;i<13;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
        for(int i=1;i<12;i++)cprintf("%c",h_t); cprintf("%c",fs_t);
        for(int i=1;i<16;i++)cprintf("%c",h_t); cprintf("%c",v);
        cout<<endl;

        file.read((char*)&d, sizeof(Dividend));
        while(!file.eof())
        {
            sln++;
            textcolor(LIGHTBLUE);
            cprintf("%c",v);
            for(int i=1;i<8;i++)cprintf(" "); cprintf("%c",v_t);
            for(int i=1;i<13;i++)cprintf(" "); cprintf("%c",v_t);
            for(int i=1;i<12;i++)cprintf(" "); cprintf("%c",v_t);
            for(int i=1;i<16;i++)cprintf(" "); cprintf("%c",v);

            textcolor(YELLOW);gotoxy(4,wherey());cprintf("%d",sln);

            textcolor(YELLOW);gotoxy(10,wherey());cprintf("%s",d.Get_name());

            textcolor(YELLOW);gotoxy(24,wherey());cprintf("%s",d.Get_date());

```

```
textcolor(LIGHTGREEN);gotoxy(41,wherey());cprintf("%.2f",d.Get_amount()
);
```

```
cout<<endl;
textcolor(LIGHTBLUE);
cprintf("%c",v);
for(int i=1;i<8;i++)cprintf(" "); cprintf("%c",v_t);
for(int i=1;i<13;i++)cprintf(" "); cprintf("%c",v_t);
for(int i=1;i<12;i++)cprintf(" "); cprintf("%c",v_t);
for(int i=1;i<16;i++)cprintf(" "); cprintf("%c",v);
cout<<endl;
```

```
file.read((char*)&d, sizeof(Dividend));
```

```
};
```

```
textcolor(LIGHTBLUE);
```

```
cprintf("%c",b_l);
```

```
for(int i=1;i<8;i++)cprintf("%c",h); cprintf("%c",b_s);
```

```
for(int i=1;i<13;i++)cprintf("%c",h); cprintf("%c",b_s);
```

```
for(int i=1;i<12;i++)cprintf("%c",h); cprintf("%c",b_s);
```

```
for(int i=1;i<16;i++)cprintf("%c",h); cprintf("%c",b_r);
```

```
cout<<"\n\n";
```

```
file.close();
```

```
textbackground(BLACK);
```

```
textcolor(GREEN);cprintf("Total Dividend: %.2f",
```

```
Get_dividend());
```

```
}
```

```
void Load_Settings()
```

```
{
```

```
fstream file(sett_file, ios::in);
```

```
char str[64];
```

```
file.getline(str,64);
```

```
STT=atof(str);
```

```
file.getline(str,64);
```

```
SER=atof(str);
```

```
file.getline(str,64);
```

```
cash=atof(str);
```

```
file.getline(str,64);
```

```
brokrate=atof(str);
```

```
file.close();
```

```
}
```

```
void Rewrite_Settings()
```

```
{
```

```
fstream file(sett_file, ios::out|ios::trunc);
```

```
file<<STT<<endl<<SER<<endl<<cash<<endl<<brokrate<<endl;
```

```
file.close();
```

```
}
```

```
int Check_holding(char c[])
```

```
{
```

```
Holding h;
```

```

        int flag=1;
        ifstream file(hold_file, ios::binary);
        if(file)
        {
            file.read((char*)&h, sizeof(Holding));
            while(!file.eof())
            {
                if(strcmp(h.Get_name(),c)==0){flag=0;break;}
                file.read((char*)&h, sizeof(Holding));
            };
        }
        file.close();
        return flag;
    }
};

#endif

```

4. INTERFACE.H

```

#ifndef INTERFACE
#define INTERFACE

#include "portfolio.h"

//~~~~~MENU~~~~~

void Create_Portfolio();
void View_Portfolio(char *title);
void Delete_Portfolio();
void Run_dividend(Portfolio &p);
void Backup_Portfolio(char *title, int flg=0);
void Restore_Portfolio(char *title);

void Run_program()
{
    int option;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Main Menu");
        cout<<"\n\n";
        textcolor(LIGHTCYAN);
        printf("1.Create new Portfolio");cout<<"\n";
        printf("2.Select Portfolio");cout<<"\n";
        printf("3.Delete Portfolio");cout<<"\n\n";
    }
}

```

```

cprintf("4.Update EOD Prices");cout<<"\n\n";
cprintf("5.Backup Portfolio");cout<<"\n";
cprintf("6.Restore Portfolio");cout<<"\n\n";
cprintf("7.ReadMe");cout<<"\n";
cprintf("8.About");cout<<"\n\n";
cprintf("0.Exit");
Enter_command(option);

clrscr();
Render_Title("Absolute Portfolio Manager");

if(option==1)
{
    Create_Portfolio();
}
else if(option==2)
{
    int menuflag=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Select Portfolio");

        textcolor(LIGHTGREEN);
        cout<<"\n\n\n\n";
        cprintf("Enter the Portfolio Name: ");gets(title);
        strupr(title);
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(menuflag);
    }while (menuflag==2);

    if(menuflag==1)
    {
        View_Portfolio(title);
        Backup_Portfolio(title,1);
    }
}
else if(option==3)
{
    Delete_Portfolio();
}
else if(option==4)
{
    clrscr();
    Render_Title("Absolute Portfolio Manager");
    Download_Bhavcopy();
    textcolor(LIGHTGREEN);
    cout<<"\n\n\n";
    cprintf("Press any key to continue...");
    getch();
}

```



```

else if(option==5)
{
    int menuflag=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Select Portfolio");

        textcolor(LIGHTGREEN);
        cout<<"\n\n\n\n";
        cprintf("Enter the Portfolio Name: ");gets(title);
        strupr(title);
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(menuflag);
    }while(menuflag==2);

    if(menuflag==1)
    {
        Backup_Portfolio(title);
    }
}
else if (option==6)
{
    int menuflag=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Select Portfolio");

        textcolor(LIGHTGREEN);
        cout<<"\n\n\n\n";
        cprintf("Enter the Portfolio Name: ");gets(title);
        strupr(title);
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(menuflag);
    }while(menuflag==2);

    if(menuflag==1)
    {
        Restore_Portfolio(title);
    }
}
else if(option==7)
{
    system("start Readme\\ReadMe.htm");
}

```

```

else if(option==8)
{
    Render_Title("Absolute Portfolio Manager");
    cout<<"\n\n\n";
    textcolor(LIGHTGREEN);

    cprintf("It is a DOS32 application that helps one to
manage his portfolio containing");
    cout<<endl;
    cprintf("the stocks traded on NSE and BSE.");cout<<endl;
    cprintf("Program created using Borland C++
v.5.02");cout<<"\n\n";

    cprintf("Created By :");cout<<"\n\t";
    cprintf("*Leaf Cutter");cout<<"\n\t";
    cprintf("*Pranav");cout<<"\n\n";
    cprintf("Special Thanks to the programmers of Curl & Unzip and
Mr. SHT for");
    cout<<endl;
    cprintf("his technical assistance.");
    getch();
}
else if(option==0)
{
    clrscr();
    Render_Title("Absolute Portfolio Manager");
    char q;
    textcolor(LIGHTRED);
    cout<<"\n\n\n\n";cprintf("Do you really wish to exit?(Y/N):
");
    cin>>q;
    if(q=='y' || q=='Y')option=0;
    else option=-1;
}
}while(option!=0);
}

void Backup_Portfolio(char *title, int flg)
{
    int interflag=0;
    char add[64]="Data\\";
    char backup_add[64];
    char backup_hol[64];
    strcat(add,title);
    strcpy(port_add, add);
    strcpy(hold_add, add);

    strcat(port_add,"_t.dat");
    strcat(hold_add,"_h.dat");

    fstream f1(port_add, ios::binary|ios::nocreate);
    fstream f2(hold_add, ios::binary|ios::nocreate);

    if(!f1 && !f2)interflag=1;
    f1.close();
    f2.close();
}

```

```

if(sizeof(file(port_add))!=0 && sizeof(file(hold_add))interflag=0;

if(interflag==0)
{
    strcpy(backup_add,port_add);
    strcat(backup_add, ".bak");
    strcpy(backup_hol,hold_add);
    strcat(backup_hol, ".bak");
    Transaction t;
    Holding h;
    ifstream file(port_add,ios::binary);
    ifstream fileh(hold_add,ios::binary);
    ofstream fob(backup_add,ios::binary);
    ofstream fobh(backup_hol,ios::binary);
    file.read((char*)&t,sizeof(Transaction));
    while(!file.eof())
    {
        fob.write((char*)&t,sizeof(Transaction));
        file.read((char*)&t,sizeof(Transaction));
    }
    fileh.read((char*)&h,sizeof(Holding));
    while(!fileh.eof())
    {
        fobh.write((char*)&h,sizeof(Holding));
        fileh.read((char*)&h,sizeof(Holding));
    }
    file.close();
    fileh.close();
    fob.close();
    fobh.close();
    if(flg==0)
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");cout<<"\n\n\n";
        cprintf("Backup Created!!");
        getch();
    }
}
else
{
    if(flg==0)
    {
        cout<<"\n\n";cprintf("Portfolio not found!!");
        cout<<"\n\n";cprintf("Press a key to return to main menu");
        getch();
    }
}
}

void Restore_Portfolio(char *title)
{
    int interflag=0;
    char add[64]="Data\\";
    char backup_add[64];
    char backup_hol[64];

```

```

    strcat(add,title);
    strcpy(port_add, add);
    strcpy(hold_add, add);

    strcat(port_add,"_t.dat");
    strcat(hold_add,"_h.dat");

    fstream f1(port_add, ios::binary|ios::nocreate);
    fstream f2(hold_add, ios::binary|ios::nocreate);

    if(!f1 && !f2)interflag=1;
    f1.close();
    f2.close();

    if(sizeoffile(port_add)!=0 && sizeoffile(hold_add))interflag=0;

    if(interflag==0)
    {
        strcpy(backup_add,port_add);
        strcat(backup_add,".bak");
        strcpy(backup_hol,hold_add);
        strcat(backup_hol,".bak");
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        char q;
        textcolor(LIGHTRED);
        cout<<"\n\n\n\n";cprintf("Do you really wish to restore this
portfolio?(Y/N): ");
        cin>>q;
        if(q=='Y' || q=='Y')
        {
            remove(port_add);
            remove(hold_add);
            rename(backup_add,port_add);
            rename(backup_hol,hold_add);
            clrscr();
            Render_Title("Absolute Portfolio Manager");cout<<"\n\n\n\n";
            cprintf("Portfolio Restored!!");
            getch();
        }
    }
    else
    {
        cout<<"\n\n\n";cprintf("Portfolio not found!!");
        cout<<"\n\n\n";cprintf("Press a key to return to main menu");
        getch();
    }
}

void Create_Portfolio()
{
    int menuflag=0;
    do
    {
        clrscr();

```

```

Render_Title("Absolute Portfolio Manager");
Render_Title("New Portfolio");

textcolor(LIGHTRED);
cout<<"\n\n\n\n";
cprintf("Enter the Portfolio Name: ");gets(title);
strupr(title);
cout<<"\n\n\n";
textcolor(LIGHTCYAN);
cprintf("1.Continue");cout<<endl;
cprintf("2.Re-enter Data");cout<<endl;
cprintf("0.Cancel");
Enter_command(menuflag);
}while(menuflag==2);

if(menuflag==1)
{
    char add[64]="Data\\";
    strcat(add,title);
    strcpy(port_add, add);
    strcpy(hold_add, add);
    strcpy(divi_add, add);
    strcpy(sett_add, add);

    strcat(port_add,"_t.dat");
    strcat(hold_add,"_h.dat");
    strcat(divi_add,"_d.dat");
    strcat(sett_add,"_s.txt");

    fstream f1(port_add, ios::binary|ios::out|ios::trunc);
    fstream f2(hold_add, ios::binary|ios::out|ios::trunc);
    fstream f3(divi_add, ios::binary|ios::out|ios::trunc);
    fstream f4(sett_add, ios::out|ios::trunc);

    f1.close();
    f2.close();
    f3.close();
    f4.close();
    Create_defaultSettings(sett_add);
    View_Portfolio(title);
}
}

void Delete_Portfolio()
{
    int menuflag=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Delete Portfolio");

        textcolor(LIGHTGREEN);
        cout<<"\n\n\n\n";
        cprintf("Enter the Portfolio Name to delete: ");gets(title);

```

```

   strupr(title);
    cout<<"\n\n\n";
    textcolor(LIGHTCYAN);
    cprintf("1.Continue");cout<<endl;
    cprintf("2.Re-enter Data");cout<<endl;
    cprintf("0.Cancel");
    Enter_command(menuflag);
    }while(menuflag==2);

if(menuflag==1)
{
    int interflag=0;
    char add[64]="Data\\";
    strcat(add,title);
    strcpy(port_add, add);
    strcpy(hold_add, add);
        strcpy(divi_add, add);
    strcpy(sett_add, add);

    strcat(port_add,"_t.dat");
    strcat(hold_add,"_h.dat");
    strcat(divi_add,"_d.dat");
    strcat(sett_add,"_s.txt");

    fstream f1(port_add, ios::binary|ios::nocreate);
    fstream f2(hold_add, ios::binary|ios::nocreate);
    fstream f3(divi_add, ios::binary|ios::nocreate);
    fstream f4(sett_add, ios::nocreate);

    if(!f1 && !f2 && !f3 && !f4)interflag=1;
    f1.close();
    f2.close();
    f3.close();
    f4.close();
    if(sizeoffile(port_add)!=0 && sizeoffile(hold_add)!=0 &&
sizeoffile(divi_add)!=0)interflag=0;

    clrscr();
    Render_Title("Absolute Portfolio Manager");
    Render_Title("Delete Portfolio");
    cout<<"\n\n\n\n";
    textcolor(LIGHTRED);
    if(interflag==0)
    {
        char ch;
        cout<<"\n\n";
        cprintf("Are you sure you want to DELETE portfolio?(Y/N): ");
        cin>>ch;
        if(ch=='y' || ch=='Y')
        {
            remove(port_add);
            remove(hold_add);
            remove(divi_add);
            remove(sett_add);
            cout<<"\n\n";cprintf("Portfolio deleted!!");
            cout<<"\n\n";cprintf("Press a key to return to main menu");

```

```

        getch();
    }
}
else
{
    cout<<"\n\n";cprintf("Portfolio not found!!");
    cout<<"\n\n";cprintf("Press a key to return to main menu");
    getch();
}
}
}

void View_Portfolio(char *title)
{
    int interflag=0;
    char add[64]="Data\\";
    strcat(add,title);
    strcpy(port_add, add);
    strcpy(hold_add, add);
    strcpy(divi_add, add);
    strcpy(sett_add, add);

    strcat(port_add,"_t.dat");
    strcat(hold_add,"_h.dat");
    strcat(divi_add,"_d.dat");
    strcat(sett_add,"_s.txt");

    fstream f1(port_add, ios::binary|ios::nocreate);
    fstream f2(hold_add, ios::binary|ios::nocreate);
    fstream f3(divi_add, ios::binary|ios::nocreate);
    fstream f4(sett_add, ios::nocreate);

    if(!f1 && !f2 && !f3 && !f4)interflag=1;
    f1.close();
    f2.close();
    f3.close();
    f4.close();
    if(sizeoffile(port_add)!=0 &&
sizeoffile(hold_add)!=0)interflag=0;

    if(interflag==0)
    {
        int interflag1=0;
        do
        {
            Portfolio port(port_add, hold_add,divi_add,sett_add);
            port.Load_Settings();
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            Render_Title(title);
            port.Output_portfolio();

            textcolor(LIGHTCYAN);
            cout<<"\n\n\n";
            cprintf("1.Add Transaction");cout<<endl;
            cprintf("2.View Transaction History");cout<<endl;

```

```

cprintf("3.Dividend");cout<<endl;
cprintf("4.View Portfolio Percentage");cout<<endl;
cprintf("5.Settings");cout<<endl;
cprintf("0.Back to Main Menu");
Enter_command(interflag1);

if(interflag1==1)
{
    Transaction t;
    int interflag2;

    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("New Transaction");
        cout<<"\n\n";
        cprintf("Transaction Type");cout<<"\t";cprintf("Stock
Exchange");cout<<"\n";
        textcolor(LIGHTGREEN);

        cprintf("1.Buy");cout<<"\t\t\t";cprintf("1.NSE");cout<<endl;

        cprintf("2.Sell");cout<<"\t\t\t";cprintf("2.BSE");cout<<endl;
        cprintf("3.Bonus");cout<<endl;
        cprintf("4.Split");cout<<endl;
        t.Input();
        textcolor(LIGHTCYAN);
        cout<<"\n\n";
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag2);
    }while(interflag2==2);

    if(interflag2==1)
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("New Transaction");
        textcolor(LIGHTRED);
        int errorflag=0;
        if(t.get_type()<1 || t.get_type()>4){cprintf("Input
Error: Incorrect TRANSACTION TYPE entered");cout<<endl;errorflag++;}
        if(t.get_exchange()!=1 &&
t.get_exchange()!=2){cprintf("Input Error: Incorrect STOCK EXCHANGE
entered");cout<<endl;errorflag++;}
        if(t.get_type()==3 || t.get_type()==4)
            if(port.Check_holding(t.get_name())!=0)
                {cprintf("Input Error: Entered Scrip name is
not a current holding");cout<<endl;errorflag++;}

        if(port.Get_cmp_n(t.get_name(),5)==0 &&
port.Get_cmp_b(t.get_name(),7)==0)
            {cprintf("Input Error: Entered
Scrip name does not exist");cout<<endl;errorflag++;}
    }
}

```



```

if(Date_formatcheck(t.get_date())==false){cprintf("Input Error: Date
has incorrect format");cout<<endl;errorflag++;}

        if(t.get_type()==1 &&
t.get_total()>cash){cprintf("Input Error: Insufficient
Balance");cout<<endl;errorflag++;}
        if(errorflag==0)port.Input(t);
        else
        {
            cout<<"\n\n\n";
            cprintf("Press any key to continue");
            getch();
        }
    }
}

else if(interflag1==2)
{
    char name[64];
    int interflag2=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Transaction History");
        cout<<"\n\n\n\n";
        textcolor(LIGHTGREEN);
        cprintf("Enter the scrip name: ");
        gets(name);
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag2);
    }while(interflag2==2);

    if(interflag2==1)
    {
        int interflag3=0;
        int total_tr=0;
        strupr(name);
        do
        {
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            Render_Title("Transaction History");
            cout<<"\n\n";
            total_tr=port.Output_transaction_history(name);
            if(total_tr==0)
            {
                clrscr();
                Render_Title("Absolute Portfolio Manager");
                Render_Title("Transaction History");
                cout<<"\n\n\n\n";
            }
        }
    }
}

```

```

        textcolor(LIGHTRED);
        cprintf("Transaction History unavailable for
<%s>!!", name);

        cout<<"\n\n";
        cprintf("Press any key to continue...");
        getch();
        interflag3=0;
    }
    else
    {
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Edit");cout<<endl;
        cprintf("2.Delete");cout<<endl;
        cprintf("0.Back");
        Enter_command(interflag3);
    }

    if(interflag3==1 && total_tr>0)
    {
        Transaction t;
        int sln, interflag4=0;
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Transaction History");
        cout<<"\n\n";
        port.Output_transaction_history(name);
        cout<<"\n\n\n";

        textcolor(LIGHTGREEN);
        cprintf("Enter the serial number(as above): ");
        cin>>sln;

        if(sln>0 && sln<=total_tr)
        {
            do
            {
                clrscr();
                Render_Title("Absolute Portfolio
Manager");

                Render_Title("Transaction History");
                cprintf("Transaction
Type");cout<<"\t";cprintf("Stock Exchange");cout<<"\n";
                textcolor(LIGHTGREEN);

                cprintf("1.Buy");cout<<"\t\t\t";cprintf("1.NSE");cout<<endl;

                cprintf("2.Sell");cout<<"\t\t\t";cprintf("2.BSE");cout<<endl;
                cprintf("3.Bonus");cout<<endl;
                cprintf("4.Split");cout<<endl;
                t.Input();
                cout<<"\n\n\n";
                textcolor(LIGHTCYAN);
                cprintf("1.Continue");cout<<endl;
                cprintf("2.Re-enter Data");cout<<endl;
                cprintf("0.Cancel");

```

```

        Enter_command(interflag4);
    }while(interflag4==2);

    if(interflag4==1)
    {
        clrscr();
        Render_Title("Absolute Portfolio
Manager");

        Render_Title("Transaction History");
        textcolor(LIGHTRED);
        int errorflag=0;
        if(t.get_type()<1 ||
t.get_type()>4){cprintf("Input Error: Incorrect TRANSACTION TYPE
entered");cout<<endl;errorflag++;}
        if(t.get_exchange() !=1 &&
t.get_exchange() !=2){cprintf("Input Error: Incorrect STOCK EXCHANGE
entered");cout<<endl;errorflag++;}
        if(t.get_type()==3 || t.get_type()==4)
        if(port.Check_holding(t.get_name()) !=0)
        {cprintf("Input Error: Entered Scrip
name is not a current holding");cout<<endl;errorflag++;}

        if(port.Get_cmp_n(t.get_name(),5)==0 &&
port.Get_cmp_b(t.get_name(),7)==0)
        {cprintf("Input Error: Entered
Scrip name does not exist");cout<<endl;errorflag++;}

        if(Date_formatcheck(t.get_date())==false){cprintf("Input Error: Date
has incorrect format");cout<<endl;errorflag++;}

    if(errorflag==0)port.Replace_transaction(t,name,sln);
        else
        {
            cout<<"\n\n\n";
            cprintf("Press any key to continue");
            getch();
        }
    }

    else
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Transaction History");
        textcolor(LIGHTRED);
        cout<<"\n\n\n\n";
        cprintf("Serial number out of bounds!!");
        cout<<"\n\n";
        cprintf("Press any key to continue...");
        getch();
    }
}

if(interflag3==2 && total_tr>0)

```

```

{
    Transaction t;
    int sln;
    clrscr();
    Render_Title("Absolute Portfolio Manager");
    Render_Title("Transaction History");
    cout<<"\n\n";
    port.Output_transaction_history(name);
    cout<<"\n\n\n";

    textcolor(LIGHTGREEN);
    cprintf("Enter the serial number(as above): ");
    cin>>sln;

    if(sln>0 && sln<=total_tr)
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Transaction History");
        cout<<"\n\n";

        textcolor(LIGHTRED);
        char ch;
        cout<<"\n\n";
        cprintf("Are you sure you want to DELETE
transaction?(Y/N): ");
        cin>>ch;
        if(ch=='y' || ch=='Y')
        {
            int tiny_flag;
            tiny_flag=port.Delete_transaction(name,
sln);

            if(tiny_flag==0)interflag3=0;
        }
    }

    else
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Transaction History");
        textcolor(LIGHTRED);
        cout<<"\n\n\n\n";
        cprintf("Serial number out of bounds!!");
        cout<<"\n\n";
        cprintf("Press any key to continue...");
        getch();
    }
}
}while(interflag3!=0);
}
else if(interflag1==3)Run_dividend(port);
else if(interflag1==4)
{
    port.Output_portfolio_pie();
}

```

```

        system("start piechart.exe");
    }
    else if(interflag1==5)
    {
        int settingflag;
        float temp;
        do
        {
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            Render_Title("Settings");
            textcolor(LIGHTGREEN);
            cout<<"\n\n\n\n";
            cprintf("1.Securities Transaction Tax");cout<<endl;
            cprintf("2.Service Tax");cout<<endl;
            cprintf("3.Brokerage Rate");cout<<endl;
            cprintf("4.Cash Deposit");cout<<endl;
            cprintf("5.Cash Withdrawal");cout<<endl;
            cprintf("0.Back");
            Enter_command(settingflag);

            if(settingflag==1)
            {
                int interflag;
                do
                {
                    clrscr();
                    Render_Title("Absolute Portfolio Manager");
                    Render_Title("Securities Transaction Tax");
                    cout<<"\n\n\n";
                    cprintf("Current STT: %.5f",STT);cout<<"\n\n";
                    cprintf("Enter the new tax rate: ");
                    cin>>temp;
                    textcolor(LIGHTCYAN);
                    cout<<"\n\n\n";
                    cprintf("1.Continue");cout<<endl;
                    cprintf("2.Re-enter Data");cout<<endl;
                    cprintf("0.Cancel");
                    Enter_command(interflag);
                }while(interflag==2);

                if(interflag==1)
                {
                    STT=temp;
                    port.Rewrite_Settings();
                }
            }

            else if(settingflag==2)
            {
                int interflag;
                do
                {
                    clrscr();
                    Render_Title("Absolute Portfolio Manager");
                    Render_Title("Service Tax");

```

```

        cout<<"\n\n\n";
        cprintf("Current SER: %.5f",SER);cout<<"\n\n";
        cprintf("Enter the new tax rate:  ");
        cin>>temp;
        textcolor(LIGHTCYAN);
        cout<<"\n\n\n";
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag);
    }while(interflag==2);

    if(interflag==1)
    {
        SER=temp;
        port.Rewrite_Settings();
    }
}

else if(settingflag==3)
{
    int interflag;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Brokerage Rate");
        cout<<"\n\n\n";
        cprintf("Current Brokerage rate:
%.5f",brokrate);cout<<"\n\n";
        cprintf("Enter the new brokerage rate(perc):
");

        cin>>temp;
        textcolor(LIGHTCYAN);
        cout<<"\n\n\n";
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag);
    }while(interflag==2);

    if(interflag==1)
    {
        brokrate=temp;
        port.Rewrite_Settings();
    }
}

else if(settingflag==4)
{
    int interflag,temp;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Cash Deposit");

```

```

        cout<<"\n\n\n";
        cprintf("Current Deposit:
%.5f",cash);cout<<"\n\n";
        cprintf("Enter the amount you wish to deposit:
");

        cin>>temp;
        textcolor(LIGHTCYAN);
        cout<<"\n\n\n";
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag);
    }while(interflag==2);

    if(interflag==1)
    {
        cash+=temp;
        port.Rewrite_Settings();
    }

    else if(settingflag==5)
    {
        int interflag,temp;
        do
        {
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            Render_Title("Cash Withdrawal");
            cout<<"\n\n\n";
            cprintf("Current Deposit:
%.5f",cash);cout<<"\n\n";
            cprintf("Enter the amount you wish to withdraw:
");

            cin>>temp;
            textcolor(LIGHTCYAN);
            cout<<"\n\n\n";
            cprintf("1.Continue");cout<<endl;
            cprintf("2.Re-enter Data");cout<<endl;
            cprintf("0.Cancel");
            Enter_command(interflag);
        }while(interflag==2);

        if(interflag==1)
        {
            cash+=temp;
            port.Rewrite_Settings();
            port.Load_Settings();
        }
    }

    }while(settingflag!=0);

    }
}while(interflag1!=0);
}

```

```

        else
        {
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            textcolor(LIGHTRED);
            cout<<"\n\n\n\n";cprintf("Portfolio not found!!");
            cout<<"\n\n";cprintf("Press a key to return to main menu");
            getch();
        }
    }

void Run_dividend(Portfolio &p)
{
    int menuflag=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividends");
        textcolor(LIGHTCYAN);
        cout<<"\n\n\n";
        cprintf("1.Add Dividend");cout<<endl;
        cprintf("2.View Dividend History(scrip)");cout<<endl;
        cprintf("3.View Dividend History(time period)");cout<<endl;
        cprintf("4.View All Dividends");cout<<endl;
        cprintf("0.Back");
        Enter_command(menuflag);

        if(menuflag==1)
        {
            Dividend div;
            int interflag=0;
            do
            {
                clrscr();
                Render_Title("Absolute Portfolio Manager");
                Render_Title("New Dividend");
                textcolor(GREEN);
                cprintf("Enter the details for new Dividend: ");
                div.Input();
                cout<<"\n\n\n";
                textcolor(LIGHTCYAN);
                cprintf("1.Continue");cout<<endl;
                cprintf("2.Re-enter Data");cout<<endl;
                cprintf("0.Cancel");
                Enter_command(interflag);
            }while(interflag==2);

            if(interflag==1)
            {
                int errorflag=0;
                clrscr();
                Render_Title("Absolute Portfolio Manager");
                Render_Title("New Dividend");
            }
        }
    }
}

```



```

        textcolor(LIGHTRED);
        if(p.Get_cmp_n(div.Get_name(),5)==0 &&
p.Get_cmp_b(div.Get_name(),7)==0){cprintf("Input Error: Entered Scrip
name does not exist");cout<<endl;errorflag++;}
        if(Date_formatcheck(div.Get_date())==false){cprintf("Input
Error: Date has incorrect format");cout<<endl;errorflag++;}

        if(errorflag==0)p.Input(div);
        else
        {
            cout<<"\n\n\n";
            cprintf("Press any key to continue");
            getch();
        }
    }
}

else if(menuflag==2)
{
    char name[64];
    int interflag=0;
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        cout<<"\n\n\n\n";
        textcolor(LIGHTGREEN);
        cprintf("Enter the scrip name: ");
        gets(name);
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag);
    }while(interflag==2);

    if(interflag==1)
    {
        int interflag1=0;
        int total_div=0;
        strupr(name);
        do
        {
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            Render_Title("Dividend History");
            cout<<"\n\n\n";
            total_div=p.Output_dividend_history(name);
            if(total_div==0)
            {
                clrscr();
                Render_Title("Absolute Portfolio Manager");
                Render_Title("Dividend History");
                cout<<"\n\n\n\n";
            }
        }
    }
}

```

```

        textcolor(LIGHTRED);
        cprintf("Dividend History unavailable for <%s>!!",
name);

        cout<<"\n\n";
        cprintf("Press any key to continue...");
        getch();
        interflag1=0;
    }
    else
    {
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Edit");cout<<endl;
        cprintf("2.Delete");cout<<endl;
        cprintf("0.Back");
        Enter_command(interflag1);
    }

    if(interflag1==1 && total_div>0)
    {
        Dividend div;
        int sln, interflag2=0;
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        cout<<"\n\n";
        p.Output_dividend_history(name);
        cout<<"\n\n\n";

        textcolor(LIGHTGREEN);
        cprintf("Enter the serial number(as above): ");
        cin>>sln;

        if(sln>0 && sln<=total_div)
        {
            do
            {
                clrscr();
                Render_Title("Absolute Portfolio Manager");
                Render_Title("Dividend History");
                textcolor(GREEN);
                cprintf("Enter the details for new Dividend:
");

                div.Input();
                cout<<"\n\n\n";
                textcolor(LIGHTCYAN);
                cprintf("1.Continue");cout<<endl;
                cprintf("2.Re-enter Data");cout<<endl;
                cprintf("0.Cancel");
                Enter_command(interflag2);
            }while(interflag2==2);

            if(interflag2==1)
            {
                int errorflag=0;
                clrscr();

```

```

        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        textcolor(LIGHTRED);
        if(p.Get_cmp_n(div.Get_name(),5)==0 &&
p.Get_cmp_b(div.Get_name(),7)==0){cprintf("Input Error: Entered Scrip
name does not exist");cout<<endl;errorflag++;}

if(Date_formatcheck(div.Get_date())==false){cprintf("Input Error: Date
has incorrect format");cout<<endl;errorflag++;}

if(errorflag==0)p.Replace_dividend(div,name,sln);
        else
        {
            cout<<"\n\n\n";
            cprintf("Press any key to continue");
            getch();
        }
    }

    else
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        textcolor(LIGHTRED);
        cout<<"\n\n\n\n";
        cprintf("Serial number out of bounds!!");
        cout<<"\n\n";
        cprintf("Press any key to continue...");
        getch();
    }
}

if(interflag1==2 && total_div>0)
{
    Dividend div;
    int sln;
    clrscr();
    Render_Title("Absolute Portfolio Manager");
    Render_Title("Dividend History");
    cout<<"\n\n";
    p.Output_dividend_history(name);
    cout<<"\n\n\n";

    textcolor(LIGHTGREEN);
    cprintf("Enter the serial number(as above): ");
    cin>>sln;

    if(sln>0 && sln<=total_div)
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        cout<<"\n\n";
    }
}

```

```

        textcolor(LIGHTRED);
        char ch;
        cout<<"\n\n";
        cprintf("Are you sure you want to DELETE
dividend?(Y/N): ");
        cin>>ch;
        if(ch=='y' || ch=='Y')
        {
            int tiny_flag;
            tiny_flag=p.Delete_dividend(name, sln);
            if(tiny_flag==0) interflag1=0;
        }
        else
        {
            clrscr();
            Render_Title("Absolute Portfolio Manager");
            Render_Title("Dividend History");
            textcolor(LIGHTRED);
            cout<<"\n\n\n\n";
            cprintf("Serial number out of bounds!!");
            cout<<"\n\n";
            cprintf("Press any key to continue...");
            getch();
        }
    }while(interflag1!=0);
}

else if(menuflag==4)
{
    clrscr();
    Render_Title("Absolute Portfolio Manager");
    Render_Title("All Dividends");
    cout<<"\n\n";
    p.Output_dividend_all();
    cout<<"\n\n\n\n";
    textcolor(LIGHTRED);
    cprintf("Press any key to return to previous menu...");
    getch();
}

else if(menuflag==3)
{
    int interflag2=0;
    char d1[9],d2[9];
    do
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        cout<<"\n\n\n\n\n";
        textcolor(LIGHTGREEN);

```

```

        cprintf("Enter start date(dd-mm-yy): ");
        gets(d1);
        cprintf("Enter end date(dd-mm-yy): ");
        gets(d2);
        cout<<"\n\n\n";
        textcolor(LIGHTCYAN);
        cprintf("1.Continue");cout<<endl;
        cprintf("2.Re-enter Data");cout<<endl;
        cprintf("0.Cancel");
        Enter_command(interflag2);
    }while(interflag2==2);

    if(interflag2==1)
    {
        clrscr();
        Render_Title("Absolute Portfolio Manager");
        Render_Title("Dividend History");
        cout<<"\n\n";
        p.Output_dividend_date(d1,d2);
        cout<<"\n\n\n";
        textcolor(LIGHTRED);
        cprintf("Press any key to return to previous menu...");
        getch();
    }
}

}while(menuflag!=0);

}

#endif

```

5. PIECHART.CPP

```

#include <iostream.h>
#include <fstream.h>
#include <conio.h>
#include <string.h>
#include <stdlib.h>
#include <iomanip.h>
#include <stdio.h>
#include <math.h>
#include <stdarg.h>
#include <dos.h>
#include <ctype.h>
#include <graphics.h>

int f_style[]={SOLID_FILL, HATCH_FILL, INTERLEAVE_FILL, XHATCH_FILL };

void main()
{
    int gdr=DETECT;int gmd;
    initgraph(&gdr,&gmd,"");

```

```

float inv,netinv=0;
float stangle=0;
float endangle;
char str[64];
char name[15];
char num[20];
int i,j,n=0,m=0;

rectangle(0,0,640,480);
floodfill(1,1,WHITE);

ifstream file("pietemp.txt");
file.getline(str,64);
while(!file.eof())
{
    for(i=0;str[i]!=':';i++)
    {
        name[i]=str[i];
    }
    name[i]='\0';
    i++;
    for(j=0;str[i]!='\0';i++, j++)
    {
        num[j]=str[i];
    }
    num[j]='\0';
    netinv+=atof(num);
    file.getline(str,64);
    n++;
};

file.close();

setcolor(WHITE);
settextstyle(COMPLEX_FONT, HORIZ_DIR, 1);
int col=9;
int f=0;
int x=10,y=10;
float perc;
int perc_x,perc_y;

ifstream temp("pietemp.txt");
temp.getline(str,64);
while(!temp.eof())
{
    for(i=0;str[i]!=':';i++)
    {
        name[i]=str[i];
    }
    name[i]='\0';
    i++;
    for(j=0;str[i]!='\0';i++, j++)
    {
        num[j]=str[i];
    }

```

```

    }
    num[j]='\0';

    inv=atof(num);
    perc=inv/netinv*100.0;
    if((int)perc==0)perc++;
    char percentage[4];
    itoa((int)perc, percentage, 10);

    setfillstyle(f_style[f], col);
    endangle=stangle+(inv/netinv*360.0);
    m++;
    if(m==n)endangle=360;
    //if(endangle-(int)endangle>=0.5 && endangle<=360)endangle++;
    pieslice(380,240,stangle,endangle,150);

    setcolor(BLACK);
    outtextxy(x,y,name);
    float radians=((stangle+endangle)/2.0)*3.14/180;
    perc_x=380+(cos(radians)*175);
    perc_y=240-(sin(radians)*175);
    outtextxy(perc_x, perc_y, percentage);outtextxy(perc_x+17,
perc_y,"%");
    bar(x+98, y-3, x+123, y+7);
    y+=15;
    setcolor(WHITE);
    col++;

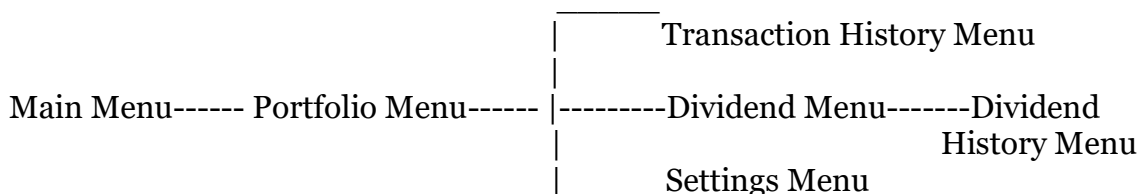
    stangle=endangle;
    temp.getline(str,64);

    if(col>=WHITE)
    {
        col=BLACK;
        if((f+1)<sizeof(f_style))f++;
    }
};
temp.close();
getch();
remove("pietemp.txt");
}

```

INTERFACE

The Menu Interface is easy to navigate and maneuver. Each command/Instruction is numbered and the user has to enter that code in the “Enter Command” space. Each Menu is linked to the previous menu from where it was accessed. The entire interface consists of 6 main interface menus:-



The Data input is handled expertly. Input errors are noted and displayed to the user. Any data if inputted wrong due to human error it can be re-entered without any problem. If the user does not wish to enter the data he may cancel the data entry providing “o” in the command interface.

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: []

In every menu “o” is provided as the universal exit/cancel command. Repeated input of “o, RETURN”, will lead to the exit conformation screen.

The basic syntax of this interface is:-

```
do
{
    Statement(s);
    Input(s);
    Input interface flag;
}while(flag!=exit);
```


OUTPUT

Absolute Portfolio Manager

~~~~~

### Main Menu

~~~~~

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio

- 4.Update EOD Prices

- 5.Backup Portfolio
- 6.Restore Portfolio

- 7.ReadMe
- 8.About

- 0.Exit

Enter Command: []

Main Menu

1. CREATING A NEW PORTFOLIO “MyPortfolio”:

a) Step I

```
Absolute Portfolio Manager
~~~~~
Main Menu
~~~~~
```

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices
- 5.Backup Portfolio
- 6.Restore Portfolio
- 7.ReadMe
- 8.About
- 0.Exit

Enter Command: [1]

//Input: 1, RETURN

b) Step II

```
Absolute Portfolio Manager
~~~~~
New Portfolio
~~~~~
```

Enter the Portfolio Name: MyPortfolio

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//Input: MyPortfolio, RETURN, 1, RETURN

c) Step III

Absolute Portfolio Manager

~~~~~ MYPORTFOLIO ~~~~~

STOCK	QUANTITY	AVG RATE	EOD PRICE	GAIN	GAIN%	LATEST VALUE

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: []

//“MyPortfolio” is now created and is automatically viewed

2. ACCESSING A PORTFOLIO

a) Step I

Absolute Portfolio Manager

~~~~~ Main Menu ~~~~~

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices
- 5.Backup Portfolio
- 6.Restore Portfolio
- 7.ReadMe
- 8.About
- 0.Exit

Enter Command: [2]

//Input: 2, RETURN

b) Step II

Absolute Portfolio Manager
~~~~~  
**Select Portfolio**  
~~~~~

Enter the Portfolio Name: MyPortfolio

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** MyPortfolio, RETURN, 1, RETURN
//But if a portfolio name is entered that does not exist then the error message is
//outputted

Absolute Portfolio Manager
~~~~~

Portfolio not found!!

Press a key to return to main menu

### c) Step III

**Absolute Portfolio Manager**  
~~~~~  
MYPORTFOLIO
~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|------|-------|--------------|
|       |          |          |           |      |       |              |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [ ]

//“MyPortfolio” is viewed

### 3. DELETING A PORTFOLIO

#### a) Step I

```
Absolute Portfolio Manager
~~~~~
Main Menu
~~~~~
```

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices
- 5.Backup Portfolio
- 6.Restore Portfolio
- 7.ReadMe
- 8.About
- 0.Exit

Enter Command: [3]

**//Input: 3, RETURN**

#### b) Step II

```
Absolute Portfolio Manager
~~~~~
Delete Portfolio
~~~~~
```

Enter the Portfolio Name to delete: MyPortfolio

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

**//Input: MyPortfolio, RETURN, 1, RETURN**

**//But if a portfolio name is entered that does not exist then the error message is  
//outputted**

```
Absolute Portfolio Manager
~~~~~
```

Portfolio not found!!

Press a key to return to main menu

c) Step III

Absolute Portfolio Manager  
~~~~~  
Delete Portfolio  
~~~~~

Are you sure you want to DELETE portfolio?(Y/N): y

Portfolio deleted!!

Press a key to return to main menu

**//Input: y, RETURN, RETURN**

4. UPDATING END OF DAY PRICES

a) Step I

Absolute Portfolio Manager  
~~~~~  
Main Menu  
~~~~~

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices
- 5.Backup Portfolio
- 6.Restore Portfolio
- 7.ReadMe
- 8.About
- 0.Exit

Enter Command: [4]

**//Input: 4, RETURN**

## Absolute Portfolio Manager

~~~~~

```
C:\Paddle\portfolio1>curl -o bhavcopy_NSEtemp.csv
http://www.nseindia.com/content/historical/EQUITIES/2007/NOV/cm19NOV2007bhav.csv
```

| % Total | % Received | % Xferd | Average Speed | Time  | Time  | Time | Current |         |         |          |      |
|---------|------------|---------|---------------|-------|-------|------|---------|---------|---------|----------|------|
|         |            |         | Dload Upload  | Total | Spent | Left | Speed   |         |         |          |      |
| 100     | 86727      | 100     | 86727         | 0     | 0     | 3867 | 0       | 0:00:22 | 0:00:22 | --:--:-- | 4790 |

```
C:\Paddle\portfolio1>curl -o bhav_bse.zip
http://www.bseindia.com/bhavcopy/eq191107_csv.zip
```

| % Total | % Received | % Xferd | Average Speed | Time  | Time  | Time | Current |         |         |          |      |
|---------|------------|---------|---------------|-------|-------|------|---------|---------|---------|----------|------|
|         |            |         | Dload Upload  | Total | Spent | Left | Speed   |         |         |          |      |
| 100     | 96180      | 100     | 96180         | 0     | 0     | 3057 | 0       | 0:00:31 | 0:00:31 | --:--:-- | 6299 |

```
C:\Paddle\portfolio1>unzip bhav_bse.zip
Archive: bhav_bse.zip
 inflating: ./EQ191107.CSV
```

```
C:\Paddle\PORTFO~2>del bhav_bse.zip
```

```
C:\Paddle\PORTFO~2>ren EQ191107.CSV bhavcopy_BSE_tmp.csv
```

Press any key to continue...

**//Input: RETURN**

**//All latest market rates will be downloaded from respective sites via internet**

## 5. MANUAL BACKUP OF PORTFOLIO

### a) Step I

## Absolute Portfolio Manager

~~~~~

### Main Menu

~~~~~

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices
- 5.Backup Portfolio
- 6.Restore Portfolio
- 7.ReadMe
- 8.About
- 0.Exit

Enter Command: [5]

**//Input: 5, RETURN**

## b) Step II

Absolute Portfolio Manager  
~~~~~  
Select Portfolio  
~~~~~

Enter the Portfolio Name: MyPortfolio

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** MyPortfolio, RETURN, 1, RETURN

//But if a portfolio name is entered that does not exist then the error message is

//outputted as above cases else The portfolio is backedup.

Absolute Portfolio Manager  
~~~~~

Backup Created!!

## 6. RESTORE OF PORTFOLIO

### a) Step I

Absolute Portfolio Manager  
~~~~~  
Main Menu  
~~~~~

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices
- 5.Backup Portfolio
- 6.Restore Portfolio
- 7.ReadMe
- 8.About
- 0.Exit

Enter Command: [6]

//**Input:** 6, RETURN



### b) Step II

Absolute Portfolio Manager  
~~~~~  
Select Portfolio  
~~~~~

Enter the Portfolio Name: MyPortfolio

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** MyPortfolio, RETURN, 1, RETURN

//But if a portfolio name is entered that does not exist then the error message is

//outputted as above cases.

### c) Step III

Absolute Portfolio Manager  
~~~~~

Do you really wish to restore this portfolio?(Y/N): y

//**Input:** y, RETURN

//The portfolio is restored from backup

Absolute Portfolio Manager  
~~~~~

Portfolio Restored!!

## 7. ACCESSING THE HELP DOCUMENTS

### a) Step I

Absolute Portfolio Manager  
~~~~~  
Main Menu  
~~~~~

- 1.Create new Portfolio
- 2.Select Portfolio
- 3.Delete Portfolio
- 4.Update EOD Prices

5.Backup Portfolio  
6.Restore Portfolio

7.ReadMe  
8.About

0.Exit

Enter Command: [7]

**//Input: 7, RETURN**

**//ReadMe document (html format) will be opened in the default application.**

## 8.ACCESSING ABOUT

### a) Step I

Absolute Portfolio Manager

~~~~~  
Main Menu  
~~~~~

1.Create new Portfolio  
2.Select Portfolio  
3.Delete Portfolio

4.Update EOD Prices

5.Backup Portfolio  
6.Restore Portfolio

7.ReadMe  
8.About

0.Exit

Enter Command: [7]

**//Input: 7, RETURN**

### b) Step II

Absolute Portfolio Manager

~~~~~  
Absolute Portfolio Manager  
~~~~~

It is a DOS32 application that helps one to manage his portfolio containing the stocks traded on NSE and BSE.  
Program created using Borland C++ v.5.02

Created By :  
\*Leaf cutter  
\*Pranav

Special Thanks to the programmers of Curl & Unzip and Mr. SHT for his technical assistance.

//About content is viewed

//**Input:** RETURN

## **8. EXITING THE APPLICATION**

### **a) Step I**

#### **Absolute Portfolio Manager**

##### **Main Menu**

1.Create new Portfolio  
2.Select Portfolio  
3.Delete Portfolio

4.Update EOD Prices

5.Backup Portfolio  
6.Restore Portfolio

7.ReadMe  
8.About

0.Exit

Enter Command: [0]

//**Input:** 0, RETURN

### **b)Step II**

#### **Absolute Portfolio Manager**

Do you really wish to exit?(Y/N): y

//**Input:** y, RETURN

//The application is terminated

## Portfolio Menu

### 1. ADDING A NEW TRANSACTION

#### a) Step I

##### Absolute Portfolio Manager

##### MYPORTFOLIO

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|------|-------|--------------|
|       |          |          |           |      |       |              |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [1]

**//Input: 1, RETURN**

#### b) Step II

##### Absolute Portfolio Manager

##### New Transaction

- |                  |                |
|------------------|----------------|
| Transaction Type | Stock Exchange |
| 1.Buy            | 1.NSE          |
| 2.Sell           | 2.BSE          |
| 3.Bonus          |                |
| 4.Split          |                |

Enter type of transaction: 1  
Enter stock exchange: 1  
Enter scrip name: rpl  
Enter quantity purchased: 10  
Enter date of purchase(dd-mm-yy): 19-11-07  
Enter purchase rate: 200

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

**//Input: 1, RETURN, 1, RETURN, rpl, RETURN, 10, RETURN, 19-11-07,  
//RETURN, 200, RETURN, 1, RETURN  
//Transaction details are entered accordingly (need not be case sensitive),  
//adhering to the legend provided.**

//Sell transaction is identical to the above. Whereas for Bonus/Split transaction a  
 //ratio stating the bonus or split is to be inputted along with the rest of the details  
 //example:

**Absolute Portfolio Manager**  
 ~~~~~  
**New Transaction**  
 ~~~~~

|                  |                |
|------------------|----------------|
| Transaction Type | Stock Exchange |
| 1.Buy            | 1.NSE          |
| 2.Sell           | 2.BSE          |
| 3.Bonus          |                |
| 4.Split          |                |

Enter type of transaction: 3  
 Enter scrip name: rpl  
 Enter bonus ratio(x:y): 1:2  
 Enter date of bonus(dd-mm-yy): 19-11-07

1.Continue  
 2.Re-enter Data  
 0.Cancel

Enter Command: [ ]

//If any input is inappropriate error messages are displayed accordingly

**Absolute Portfolio Manager**  
 ~~~~~  
**New Transaction**  
 ~~~~~

Input Error: Entered Scrip name does not exist  
 Input Error: Date has incorrect format  
 Input Error: Insufficient Balance

Press any key to continue

//Input: RETURN  
 //else

**Absolute Portfolio Manager**  
 ~~~~~  
**MYPORTFOLIO**  
 ~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN  | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|-------|-------|--------------|
| RPL   | 10       | 200.25   | 208.80    | 85.50 | 4.3   | 2088.00      |

|                       |            |        |
|-----------------------|------------|--------|
| Unrealised Gain Today | : -56.50   | -2.71% |
| Portfolio Value Today | : 2088.50  | 4.29%  |
| Balance Amount        | : 97997.50 |        |



|                       |   |          |        |
|-----------------------|---|----------|--------|
| Unrealised Gain Today | : | -178.75  | -0.26% |
| Portfolio Value Today | : | 67616.75 | 1.55%  |
| Balance Amount        | : | 33416.90 |        |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [2]

**//Input: 2, RETURN**

b) Step II

```

Absolute Portfolio Manager
~~~~~
Transaction History
~~~~~

```

Enter the scrip name: rpl

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

**//Input: rpl, RETURN, 1, RETURN**

**//Error is displayed if no transaction of entered scrip exists or the scrip does not exist. Example scrip RPLO**

```

Absolute Portfolio Manager
~~~~~
Transaction History
~~~~~

```

Transaction History unavailable for <RPLO>!!

Press any key to continue...

**//Input: RETURN**

**Absolute Portfolio Manager**  
 ~~~~~  
**Transaction History**  
 ~~~~~

Transaction History <RPL> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN  |
|-------|----------|----------|----------|-----------|-----------|-------|
| 1(N)  | 19-11-07 | 10       | 200.00   | ---       | 208.80    | 88.00 |

- 1.Edit
- 2.Delete
- 0.Back

Enter Command: [ ]

//The Transaction History for the scrip is displayed

### 3. ACCESSING DIVIDEND MENU

#### a) Step I

**Absolute Portfolio Manager**  
 ~~~~~  
**MYPORTFOLIO**  
 ~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN   | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|--------|-------|--------------|
| RPL   | 10       | 200.25   | 208.80    | 85.50  | 4.3   | 2088.00      |
| LT    | 15       | 4305.38  | 4368.50   | 946.88 | 1.5   | 65527.50     |

|                       |            |        |
|-----------------------|------------|--------|
| Unrealised Gain Today | : -178.75  | -0.26% |
| Portfolio Value Today | : 67616.75 | 1.55%  |
| Balance Amount        | : 33416.90 |        |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [3]

//Input: 3, RETURN



Absolute Portfolio Manager  
~~~~~  
Dividends  
~~~~~

- 1.Add Dividend
- 2.View Dividend History(scrip)
- 3.View Dividend History(time period)
- 4.View All Dividends
- 0.Back

Enter Command: [ ]

//Dividend Menu is accessed

**4. VIEWING PORTFOLIO PERCENTAGE**

**a) Step I**

Absolute Portfolio Manager  
~~~~~  
MYPORTFOLIO  
~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN   | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|--------|-------|--------------|
| RPL   | 10       | 200.25   | 208.80    | 85.50  | 4.3   | 2088.00      |
| LT    | 15       | 4305.38  | 4368.50   | 946.88 | 1.5   | 65527.50     |

|                       |   |          |  |        |
|-----------------------|---|----------|--|--------|
| Unrealised Gain Today | : | -178.75  |  | -0.26% |
| Portfolio Value Today | : | 67616.75 |  | 1.55%  |
| Balance Amount        | : | 33416.90 |  |        |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [4]

//**Input: 4, RETURN**

//Portfolio Percentage Pie chart is viewed drawn by dos bgi graphics

## 5. ACCESSING THE SETTINGS MENU

### a) Step I

#### Absolute Portfolio Manager

~~~~~

#### MYPORTFOLIO

~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN   | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|--------|-------|--------------|
| RPL   | 10       | 200.25   | 208.80    | 85.50  | 4.3   | 2088.00      |
| LT    | 15       | 4305.38  | 4368.50   | 946.88 | 1.5   | 65527.50     |

Unrealised Gain Today : -178.75 -0.26%  
Portfolio Value Today : 67616.75 1.55%  
Balance Amount : 33416.90

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [5]

//Input: 5, RETURN

//Settings menu is accessed

#### Absolute Portfolio Manager

~~~~~

#### Settings

~~~~~

- 1.Securities Transaction Tax
- 2.Service Tax
- 3.Brokerage Rate
- 4.Cash Deposit
- 5.Cash Withdrawal
- 0.Back

Enter Command: [ ]

## 6. EXITING TO MAIN MENU

### a) Step I

#### Absolute Portfolio Manager

#### ~~~~~ MYPORTFOLIO ~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN   | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|--------|-------|--------------|
| RPL   | 10       | 200.25   | 208.80    | 85.50  | 4.3   | 2088.00      |
| LT    | 15       | 4305.38  | 4368.50   | 946.88 | 1.5   | 65527.50     |

Unrealised Gain Today : -178.75 -0.26%  
Portfolio Value Today : 67616.75 1.55%  
Balance Amount : 33416.90

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [0]

**//Input: o, RETURN**

**//Program exits to main menu. The portfolio is auto back upped.**

## Transaction History Menu

### 1. EDITING A TRANSACTION

#### a) Step I

Absolute Portfolio Manager  
~~~~~  
Transaction History  
~~~~~

Transaction History <RPL> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN  |
|-------|----------|----------|----------|-----------|-----------|-------|
| 1(N)  | 19-11-07 | 10       | 200.00   | ---       | 208.80    | 88.00 |

1.Edit  
2.Delete  
0.Back

Enter Command: [1]

**//Input: 1, RETURN**

#### b) Step II

Absolute Portfolio Manager  
~~~~~  
Transaction History  
~~~~~

Transaction History <RPL> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN  |
|-------|----------|----------|----------|-----------|-----------|-------|
| 1(N)  | 19-11-07 | 10       | 200.00   | ---       | 208.80    | 88.00 |

Enter the serial number(as above): 1

**//Input: 1, RETURN**

#### c) Step III

Absolute Portfolio Manager  
~~~~~  
Transaction History  
~~~~~

Transaction Type  
1.Buy  
2.Sell  
3.Bonus  
4.Split

Stock Exchange  
1.NSE  
2.BSE

Enter type of transaction: 1  
 Enter stock exchange: 1  
 Enter scrip name: rpl  
 Enter quantity purchased: 10  
 Enter date of purchase(dd-mm-yy): 18-11-07  
 Enter purchase rate: 200

1.Continue  
 2.Re-enter Data  
 0.Cancel

Enter Command: [1]

//Input: 1, RETURN, 1, RETURN, rpl, RETURN, 10, RETURN, 18-11-07,  
 //RETURN, 200, RETURN, 1, RETURN  
 //The transaction will be edited accordingly and follows same rules as addition of  
 //a new transaction

**Absolute Portfolio Manager**  
 ~~~~~  
**Transaction History**  
 ~~~~~

Transaction History <RPL> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN  |
|-------|----------|----------|----------|-----------|-----------|-------|
| 1(N)  | 18-11-07 | 10       | 200.00   | ---       | 208.80    | 88.00 |

1.Edit  
 2.Delete  
 0.Back

Enter Command: [ ]

## 1. DELETE A TRANSACTION

### a) Step I

**Absolute Portfolio Manager**  
 ~~~~~  
**Transaction History**  
 ~~~~~

Transaction History <RPL> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN  |
|-------|----------|----------|----------|-----------|-----------|-------|
| 1(N)  | 18-11-07 | 10       | 200.00   | ---       | 208.80    | 88.00 |

- 1.Edit
- 2.Delete
- 0.Back

Enter Command: [2]

**//Input: 2, RETURN**

b) Step II

Absolute Portfolio Manager  
~~~~~  
Transaction History  
~~~~~

Transaction History <RPL> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN  |
|-------|----------|----------|----------|-----------|-----------|-------|
| 1(N)  | 18-11-07 | 10       | 200.00   | ---       | 208.80    | 88.00 |

Enter the serial number(as above): 1

**//Input: 1, RETURN**

c) Step III

Absolute Portfolio Manager  
~~~~~  
Transaction History  
~~~~~

Are you sure you want to DELETE transaction?(Y/N): y

**//Input: y, RETURN**

**//The transaction was deleted and program returns to portfolio menu as there weren't any more transactions from that particular scrip, else it would have returned to the transaction history menu.**

Absolute Portfolio Manager  
~~~~~  
MYPORTFOLIO  
~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN   | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|--------|-------|--------------|
| LT    | 15       | 4305.38  | 4368.50   | 946.88 | 1.5   | 65527.50     |

|                       |            |        |
|-----------------------|------------|--------|
| Unrealised Gain Today | : -122.25  | -0.19% |
| Portfolio Value Today | : 65528.25 | 1.47%  |
| Balance Amount        | : 31414.40 |        |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [ ]

### 3. EXITING TO PORTFOLIO MENU

//The transaction history of LT is entered from the portfolio menu

#### a) Step I

Absolute Portfolio Manager  
~~~~~  
Transaction History  
~~~~~

Transaction History <LT> :

| SL_NO | DATE     | QUANTITY | BUY RATE | SALE RATE | EOD PRICE | GAIN    |
|-------|----------|----------|----------|-----------|-----------|---------|
| 1(N)  | 19-11-07 | 15       | 4300.00  | ---       | 4368.50   | 1027.50 |

- 1.Edit
- 2.Delete
- 0.Back

Enter Command: [0]

//**Input:** o, RETURN//The Program returns to the Portfolio menu.

Absolute Portfolio Manager  
~~~~~  
MYPORTFOLIO  
~~~~~

| STOCK | QUANTITY | AVG RATE | EOD PRICE | GAIN   | GAIN% | LATEST VALUE |
|-------|----------|----------|-----------|--------|-------|--------------|
| LT    | 15       | 4305.38  | 4368.50   | 946.88 | 1.5   | 65527.50     |

|                       |            |        |
|-----------------------|------------|--------|
| Unrealised Gain Today | : -122.25  | -0.19% |
| Portfolio Value Today | : 65528.25 | 1.47%  |
| Balance Amount        | : 31414.40 |        |

- 1.Add Transaction
- 2.View Transaction History
- 3.Dividend
- 4.View Portfolio Percentage
- 5.Settings
- 0.Back to Main Menu

Enter Command: [ ]

## Dividend Menu

### 1. ADDING A DIVIDEND

#### a) Step I

```
Absolute Portfolio Manager
~~~~~
Dividends
~~~~~
```

- 1.Add Dividend
- 2.View Dividend History(scrip)
- 3.View Dividend History(time period)
- 4.View All Dividends
- 0.Back

Enter Command: [1]

//**Input:** 1, RETURN

#### b) Step II

```
Absolute Portfolio Manager
~~~~~
New Dividend
~~~~~
```

Enter the details for new Dividend:

```
Enter the scrip name: lt
Enter date(dd-mm-yy): 19-11-07
Enter dividend amount: 5000
```

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** lt, RETURN, 19-11-07, RETURN, 5000, RETURN, 1, RETURN

//New dividend is added.

//If any input is inappropriate error messages are displayed accordingly

```
Absolute Portfolio Manager
~~~~~
New Dividend
~~~~~
```

Input Error: Entered Scrip name does not exist  
Input Error: Date has incorrect format

Press any key to continue

//**Input:** RETURN

//else



//Another dividend from LT of amount 2000 dated 15-11-07 is added

Absolute Portfolio Manager

~~~~~  
New Dividend  
~~~~~

Enter the details for new Dividend:

Enter the scrip name: lt  
Enter date(dd-mm-yy): 15-11-07  
Enter dividend amount: 2000

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//Input: lt, RETURN, 15-11-07, RETURN, 2000, RETURN, 1, RETURN

## 2. VIEWING DIVIDEND HISTORY SCRIP WISE

### a) Step I

Absolute Portfolio Manager

~~~~~  
Dividends  
~~~~~

- 1.Add Dividend
- 2.View Dividend History(scrip)
- 3.View Dividend History(time period)
- 4.View All Dividends
- 0.Back

Enter Command: [2]

//Input: 2, RETURN

### b) Step II

Absolute Portfolio Manager

~~~~~  
Dividend History  
~~~~~

Enter the scrip name: lt

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** lt, RETURN, 2, RETURN  
//Dividend history is viewed arranged by date for the entered scrip.

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 15-11-07 | 2000.00 |
| 2     | 19-11-07 | 5000.00 |

Total Dividend: 7000.00

- 1.Edit
- 2.Delete
- 0.Back

Enter Command: [ ]

**3. VIEW DIVIDEND HISTORY TIME WISE**

**a) Step I**

Absolute Portfolio Manager  
~~~~~  
Dividends  
~~~~~

- 1.Add Dividend
- 2.View Dividend History(scrip)
- 3.View Dividend History(time period)
- 4.View All Dividends
- 0.Back

Enter Command: [3]

//**Input:** 3, RETURN

**b) Step II**

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Enter start date(dd-mm-yy): 1-11-07  
Enter end date(dd-mm-yy): 18-11-07

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** 1, RETURN

//Dividend History is viewed for all dividends between 1/11/07 to 18/11/07

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History:

| SL_NO | STOCK | DATE     | AMOUNT  |
|-------|-------|----------|---------|
| 1     | LT    | 15-11-07 | 2000.00 |

Total Dividend: 2000.00

Press any key to return to previous menu...

//**Input:** RETURN

**4. VIEW COMPLETE DIVIDEND HISTORY**

a) Step I

Absolute Portfolio Manager  
~~~~~  
Dividends  
~~~~~

- 1.Add Dividend
- 2.View Dividend History(scrip)
- 3.View Dividend History(time period)
- 4.View All Dividends
- 0.Back

Enter Command: [4]

//**Input:** 4, RETURN

//Dividend History is viewed

Absolute Portfolio Manager  
~~~~~  
All Dividends  
~~~~~

All Dividends:

| SL_NO | STOCK | DATE     | AMOUNT  |
|-------|-------|----------|---------|
| 1     | LT    | 15-11-07 | 2000.00 |
| 2     | LT    | 19-11-07 | 5000.00 |

Total Dividend: 7000.00

Press any key to return to previous menu...

//Input: RETURN

5. EXIT TO PORTFOLIO MENU

a) Step I

Absolute Portfolio Manager  
~~~~~  
Dividends  
~~~~~

- 1.Add Dividend
- 2.View Dividend History(scrip)
- 3.View Dividend History(time period)
- 4.View All Dividends
- 0.Back

Enter Command: [0]

//Input: o, RETURN

//Program exits to Portfolio menu

## Dividend History Menu

### 1. EDITING A DIVIDEND

#### a) Step I

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 15-11-07 | 2000.00 |
| 2     | 19-11-07 | 5000.00 |

Total Dividend: 7000.00

1.Edit  
2.Delete  
0.Back

Enter Command: [1]

//Input: 1, RETURN

#### b) Step II

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 15-11-07 | 2000.00 |
| 2     | 19-11-07 | 5000.00 |

Total Dividend: 7000.00

Enter the serial number(as above): 1

//Input: 1, RETURN

c) Step III

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Enter the details for new Dividend:

Enter the scrip name: lt  
Enter date(dd-mm-yy): 10-11-07  
Enter dividend amount: 1500

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** lt, RETURN, 10-11-07, RETURN, 1500, RETURN, 1, RETURN  
//the Dividend will be edited

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 10-11-07 | 1500.00 |
| 2     | 19-11-07 | 5000.00 |

Total Dividend: 6500.00

- 1.Edit
- 2.Delete
- 0.Back

Enter Command: [ ]

## 2. DELETING A DIVIDEND

### a) Step I

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 10-11-07 | 1500.00 |
| 2     | 19-11-07 | 5000.00 |

Total Dividend: 6500.00

- 1.Edit
- 2.Delete
- 0.Back

Enter Command: [2]

**//Input: 2, RETURN**

### b) Step II

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 10-11-07 | 1500.00 |
| 2     | 19-11-07 | 5000.00 |

Total Dividend: 6500.00

Enter the serial number(as above): 1

**//Input: 1, RETURN**

c) Step III

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Are you sure you want to DELETE dividend?(Y/N): y

//**Input: y, RETURN**

//the dividend will be deleted

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 19-11-07 | 5000.00 |

Total Dividend: 5000.00

1.Edit  
2.Delete  
0.Back

Enter Command: [ ]



### 3. EXITING TO DIVIDEND MENU

#### a) Step I

Absolute Portfolio Manager  
~~~~~  
Dividend History  
~~~~~

Dividend History <LT> :

| SL_NO | DATE     | AMOUNT  |
|-------|----------|---------|
| 1     | 19-11-07 | 5000.00 |

Total Dividend: 5000.00

1.Edit  
2.Delete  
0.Back

Enter Command: [0]

//**Input:** 0, RETURN

//The program exits to the dividend Menu

## Settings Menu

### 1. CHANGING THE SECURITY TRANSACTION TAX (STT)

#### a) Step I

Absolute Portfolio Manager  
~~~~~  
Settings  
~~~~~

1. Securities Transaction Tax  
2. Service Tax  
3. Brokerage Rate  
4. Cash Deposit  
5. Cash Withdrawal  
0. Back

Enter Command: [1]

//**Input:** 1, RETURN

#### b) Step II

Absolute Portfolio Manager  
~~~~~  
Securities Transaction Tax  
~~~~~

Current STT: 0.00125

Enter the new tax rate: 0.00125

1. Continue  
2. Re-enter Data  
0. Cancel

Enter Command: [1]

//**Input:** 0.00125, RETURN, 1, RETURN  
//STT changes to 0.00125

## 2. CHANGING THE SERVICE TAX

### a) Step I

Absolute Portfolio Manager  
~~~~~  
Settings  
~~~~~

- 1. Securities Transaction Tax
- 2. Service Tax
- 3. Brokerage Rate
- 4. Cash Deposit
- 5. Cash Withdrawal
- 0. Back

Enter Command: [2]

//Input: 2, RETURN

### b) Step II

Absolute Portfolio Manager  
~~~~~  
Service Tax  
~~~~~

Current SER: 0.12250

Enter the new tax rate: 0.12250

- 1. Continue
- 2. Re-enter Data
- 0. Cancel

Enter Command: [1]

//Input: 0.12250, RETURN, 1, RETURN

//STT changes to 0.12250

## 3. CHANGING THE BROKERAGE RATE

### a) Step I

Absolute Portfolio Manager  
~~~~~  
Settings  
~~~~~

- 1. Securities Transaction Tax
- 2. Service Tax
- 3. Brokerage Rate
- 4. Cash Deposit
- 5. Cash Withdrawal
- 0. Back

Enter Command: [3]

//Input: 3, RETURN

b) Step II

Absolute Portfolio Manager  
~~~~~  
Brokerage Rate  
~~~~~

Current Brokerage rate: 0.00000

Enter the new brokerage rate(perc): 0

1.Continue  
2.Re-enter Data  
0.Cancel

Enter Command: [1]

//Input: 0, RETURN, 1, RETURN  
//Brokerage rate changes to 0

#### 4. DEPOSITING CASH IN BALANCE

a) Step I

Absolute Portfolio Manager  
~~~~~  
Settings  
~~~~~

1.Securities Transaction Tax  
2.Service Tax  
3.Brokerage Rate  
4.Cash Deposit  
5.Cash withdrawal  
0.Back

Enter Command: [4]

//Input: 4, RETURN

b) Step II

Absolute Portfolio Manager  
~~~~~  
Cash Deposit  
~~~~~

Current Deposit: 31414.40039

Enter the amount you wish to deposit: 20000

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** 20000, RETURN, 1, RETURN

//Balance changes to 51414.39844

5. WITHDRAWING FROM CASH DEPOSIT

a) Step I

Absolute Portfolio Manager  
~~~~~  
Settings  
~~~~~

- 1.Securities Transaction Tax
- 2.Service Tax
- 3.Brokerage Rate
- 4.Cash Deposit
- 5.Cash Withdrawal
- 0.Back

Enter Command: [5]

//**Input:** 5, RETURN

b) Step II

Absolute Portfolio Manager  
~~~~~  
Cash Withdrawal  
~~~~~

Current Deposit: 51414.39844

Enter the amount you wish to withdraw: 0

- 1.Continue
- 2.Re-enter Data
- 0.Cancel

Enter Command: [1]

//**Input:** 0, RETURN, 1, RETURN  
//Balance changes to 51414.39844

## 5. EXITING THE PROGRAM

### a) Step I

Absolute Portfolio Manager  
~~~~~  
Settings  
~~~~~

- 1.Securities Transaction Tax
- 2.Service Tax
- 3.Brokerage Rate
- 4.Cash Deposit
- 5.Cash Withdrawal
- 0.Back

Enter Command: [0]

//**Input:** 0, RETURN  
//Program returns to Portfolio Menu