Stock Exchange

C++ Project

2015-2016

BOARD REGISTRATION NUMBER:

DONE BY:

SAURABH P BHANDARI

CERTIFICATE

This is to certify that Saurabh P Bhandari of class XII, Sindhi High School, Bangalore has successfully completed his project, Stock Exchange, for the AISSCE as prescribed by CBSE in the year 2015-2016.

Date:

Board Registration Number:

Signature of Internal Examiner:

Signature of External Examiner:

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Principal, Vice Principal and the support staff for lending school facilities for our project. We would like to thank Mrs. Rashmi B A for her tireless perseverance and help. We would also like to thank our parents and friends for their support and encouragement.

CONTENTS

SYNOPSIS

This is a stock exchange simulation software which gives you a basic idea of a stock exchange and provides a simulated experience of investing in stocks and shares. You can perform the following operations:-

Buy Shares: You can buy shares of different companies.

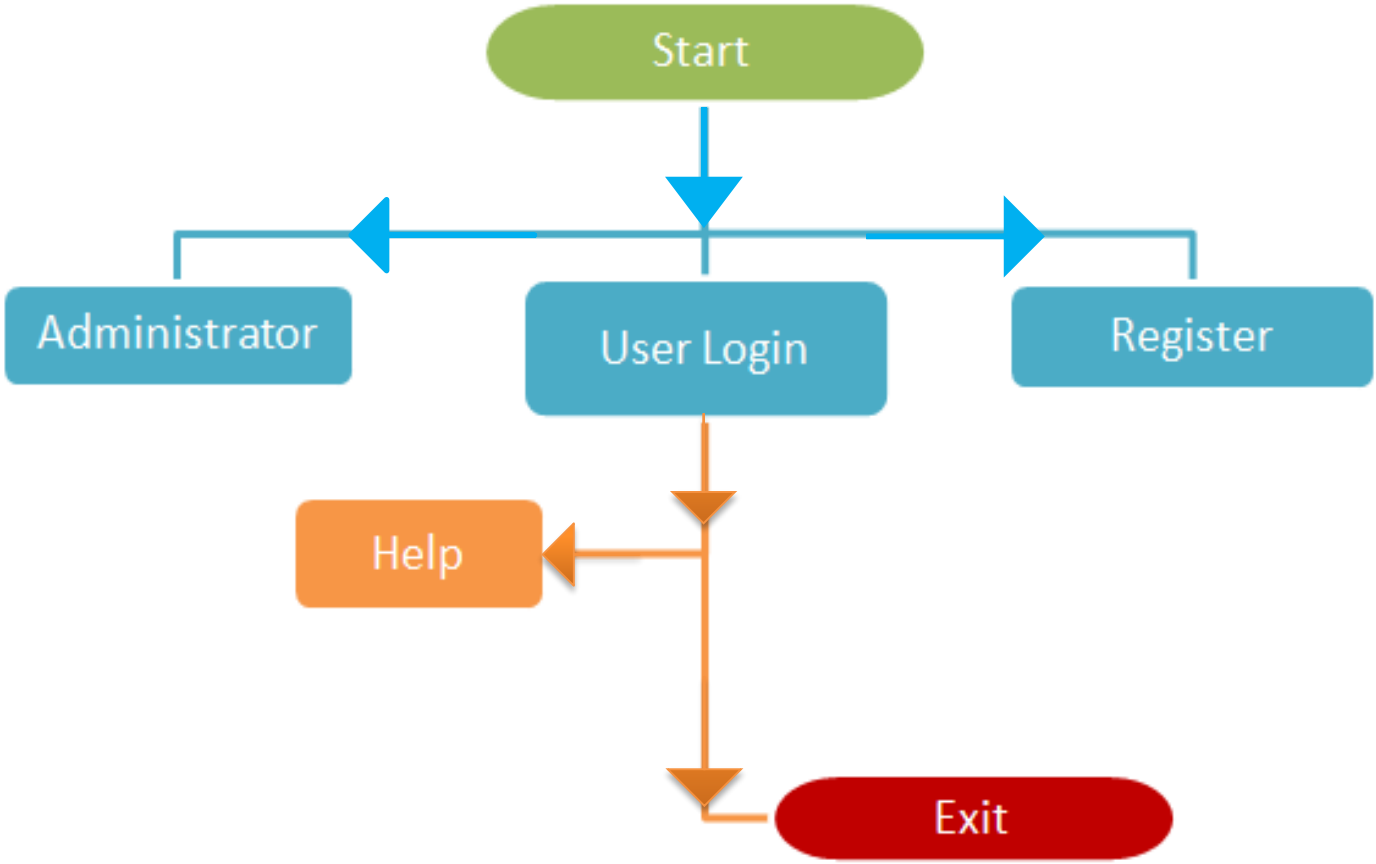
Sell Shares: You can sell the shares you own.

Other features include checking your account status, viewing the stock price-time graph of different companies, registering as a user, etc.

SYSTEM REQUIREMENTS

- INTEL CELRON(HIGHER) 333 MHz
- MS WINDOWS 95 or Higher MAC OSX-5
- 64 MB RAM
- 10 MB Hard Disk Space
- Office 2000 or Higher
- Turbo C++

FLOWCHART



DATA DICTIONARY

Header Files Used

- conio.h
- ctype.h
- dos.h
- fstream.h
- graphics.h
- iomanip.h
- iostream.h
- process.h
- stdio.h
- stdlib.h
- string.h
- time.h

Classes and Structures Used

- class **Home**
- struct **XYScatterGraph**
- class **account**
- class **shares**
- class **company**
- class **newu**

Macros

- #define qe cout<<endl;

Non-Member Functions

- void **main** ()
- void **mainl** ()
- void **admin** ()
- void **adminfnt** ()
- void **lcom** ()
- void **ucom** ()
- void **newuser** ()
- void **user** ()
- void **info** ()
- void **stock** (char[])
- void **per** (double, double)
- void **rand_double** (double, double &)
- void **portfolio** (char[])
- void **balance** (char[])
- void **buy** (char[])
- void **sell** (char[])
- void **buycheck** (char[], char[])

- void **sellcheck** (char *)
- void **upg** ()
- void **downg** ()
- void **centerstring** (char *s)
- void **intro** ()
- int **graphics** (int)
- int **initmouse** ()
- int **restrictmouseptr** (int, int, int, int)
- int **showmouseptr** ()
- int **hidemouseptr** ()
- int **getmousepos** (int *, int *, int *)
- void **stock** (char name[20])
- void **portfolio** (char n[20])
- void **balance** (char name[20])
- void **buy** (char name[20])
- void **sell** (char name[20])
- void **buycheck** (char name[20], char cname[20])

Variables

- union REGS i o

Class Home

Visibility mode: Public

Member Functions:

Function Name	Remarks
void homeI()	Displays the main menu
void box(int i,int j,int k,int l,int c,int s)	Creates a box and fills colour into it
void getchoice_home()	Mouse input for the main menu
void animate(int,int)	Animation for main menu

Private Data Members:

int button,x,y,r;

Structure XYScatterGraph

Visibility mode: Public

Member Functions:

Function Name	Remarks
XYScatterGraph(int c,int numpoints)	Accepts color and no.of points in the graph
voidFillPoints(double yarr[],double xarr[],int len)	Accepts coordinates of x and y axis
void draw()	Plots the point on the screen
void DrawGrid(int index=1)	Draws the grid for the graph

Public Data Members:

int color;

int points;

double xpoints[1000];

double ypoints[1000];

double dataxmin;

double dataxmax;

double dataymin;

double dataymax;

double xlen;

double ylen;

Class account

Visibility mode: Public

Member Functions:

Function Name	Remarks
void putdata()	Gives account related information

Public data members:

char acno[20];

char name[20];

int balance;

Class shares

Visibility mode: Public

Member functions:

Function name	Remarks
void putshares()	Displays the current profile of a company

Public data members:

int nos;

char compname[50];

double fvalue;

double cvalue;

Class company

Visibility mode: Public

Member functions:

Function name	Remarks
void getdata()	Input data for a company
void putdata()	Display data for a company
int getcode()	Returns the company code
char* getname()	Returns the company name

Public data members:

char cname[30];

int ccode;

int nshares;

double cvalue;

double fvalue;

Class newu

Visibility mode: Public

Member functions:

Function name	Remarks
void get()	Input for registering as a new user
char* getname()	Returns the name of the current user
char* getacno()	Returns the account number of the current user
void put1()	Display user menu
void put()	Display profile of the current user
void put2()	Admin function to display users list

Private data members:

char name[20];

int age;

char pan[20];

char adr[100];

char acno[20];

SOURCE CODE

```
#include<conio.h>
#include<ctype.h>
#include<dos.h>
#include<fstream.h>
#include<graphics.h>
#include<iomanip.h>
#include<iostream.h>
#include<process.h>
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<time.h>
#define qe cout<<endl;
void main();
void main1();
void admin();
void adminfnt();
void lcom();
void ucom();
void newuser();
void user();
void info();
void stock(char[]);
void per(double,double);
void rand_double(double,double&);
void portfolio(char[]);
void balance(char[]);
void buy(char[]);
void sell(char[]);
void buycheck(char[],char[]);
void sellcheck(char*);
void upg();
void downg();
void centerstring(char*s);
void intro();
int graphics(int);
int initmouse();
int restrictmouseptr(int,int,int,int);
int showmouseptr();
int hidemouseptr();
int getmousepos(int*,int*,int*);
union REGS i, o;
static const int LTMARGIN = 60;
static const int TPMARGIN = 100;
class Home
{
private:
    int button,x,y,r;
    void animate(int,int);
public:
    Home()
    {
```

```
        r=1;
    }
    void home1();
    void box(int i,int j,int k,int l,int=0,int=0);
    void getchoice_home();
};
void Home::box(int i,int j,int k,int l,int c,int s)
{
    setcolor(WHITE);
    line(i,j,k,j);
    line(i,j,i,l);
    setcolor(DARKGRAY);
    line(i,l,k,l);
    line(k,j,k,l);
    if(s!=0)
    {
        rectangle(i+1,j+1,k-1,l-1);
        setfillstyle(1,c);
        floodfill(i+2,j+2,DARKGRAY);
    }
}
void Home::home1()
{
    cleardevice();
    setbkcolor(RED);
    setfillstyle(1,RED);
    setcolor(DARKGRAY);
    rectangle(0,1,639,479);
    floodfill(2,2,DARKGRAY);
    setfillstyle(1,BROWN);
    box(230,90,410,120+1,6,1);
    box(230,315,410,345+1,6,1);
    box(230,200,410,230+1,6,1);
    box(400,400,570,430+1,6,1);
    box(100,400,270,430+1,6,1);
    setcolor(YELLOW);
    settextstyle(2,HORIZ_DIR,5);
    outtextxy(237,97,"  * ADMINISTRATOR *");
    outtextxy(247,207,"  * USER LOGIN *");
    outtextxy(257,323,"  * REGISTER *");
    outtextxy(422,407,"    * HELP * ");
    outtextxy(122,407,"    * EXIT * ");
    setfillstyle(1,BLUE);
    setcolor(BLUE);
    rectangle(1,1,638,70);
    rectangle(4,4,635,67);
    floodfill(2,2,BLUE);
    rectangle(1,450,638,479);
    floodfill(2,452,BLUE);
    rectangle(4,454,635,476);
    rectangle(1,1,638,479);
    setfillstyle(2,BLUE);
    setcolor(BLUE);
    rectangle(1,70,30,450);
    floodfill(2,72,BLUE);
    rectangle(610,70,638,450);
    floodfill(612,72,BLUE);
    setfillstyle(1,BLUE);
```

```
    floodfill(6,6,BLUE);
    setcolor(YELLOW);
    settextstyle(0,HORIZ_DIR,3);
    outtextxy(5,25," Stock Exchange 20.0");
    setcolor(YELLOW);
    settextstyle(2,HORIZ_DIR,5);
    outtextxy(17,457,"Project developed by:- Saurabh P Bhandari and
Pallab Ray");
}
void Home::animate(int n,int m)
{
    r+=2;
    if(r>10||r<1)r=1;
    if(n==1)
    {
        setcolor(YELLOW);
        switch(m)
        {
        case 1 :
            rectangle(235,93,405,117);
            setcolor(YELLOW);
            circle(200,101,r);
            break;
        case 4 :
            rectangle(235,203,405,227);
            setcolor(YELLOW);
            circle(200,215,r);
            break;
        case 7 :
            rectangle(235,317,405,342);
            setcolor(YELLOW);
            circle(200,332,r);
            break;
        case 8 :
            rectangle(105,403,265,427);
            setcolor(YELLOW);
            circle(70,411,r);
            break;
        case 9 :
            rectangle(405,403,565,427);
            setcolor(YELLOW);
            circle(370,411,r);
            break;
        }
        setcolor(CYAN);
        if(m!=1)rectangle(235,93,405,117);
        if(m!=4)rectangle(235,203,405,227);
        if(m!=7)rectangle(235,318,405,342);
        if(m!=8)rectangle(105,403,265,427);
        if(m!=9)rectangle(405,403,565,427);
        setcolor(0);
        delay(100);
        circle(200,101,r);
        circle(200,215,r);
        circle(200,332,r);
        circle(70,411,r);
        circle(370,411,r);
    }
```

```
    if(n==2)
    {
        setcolor(CYAN);
        rectangle(235,93,405,117);
        rectangle(235,203,405,227);
        rectangle(235,318,405,342);
        rectangle(105,403,265,427);
        rectangle(405,403,565,427);
        setcolor(0);
        circle(200,101,r);
        circle(200,215,r);
        circle(200,332,r);
        circle(70,411,r);
        circle(370,411,r);
    }
}
void Home::getchoice_home()
{
    home1();
    initmouse();
    showmouseptr();
    while(1)
    {
        getmousepos(&button,&x,&y);
        if(x>220&&x<410&&y>80&&y<120)animate(1,1);
        else if(x>90&&x<270&&y>140&&y<180)animate(1,2);
        else if(x>390&&x<570&&y>140&&y<180)animate(1,3);
        else if(x>220&&x<410&&y>200&&y<240)animate(1,4);
        else if(x>90&&x<270&&y>260&&y<300)animate(1,5);
        else if(x>390&&x<570&&y>260&&y<300)animate(1,6);
        else if(x>220&&x<410&&y>330&&y<370)animate(1,7);
        else if(x>90&&x<270&&y>390&&y<430)animate(1,8);
        else if(x>390&&x<570&&y>390&&y<430)animate(1,9);
        else animate(2,0);
        if(x>230&&x<410&&y>90&&y<120&&(button &1)==1)
        {
            delay(200);
            cleardevice();
            closegraph();
            admin();
        }
        if(x>220&&x<410&&y>200&&y<240&&(button &1)==1)
        {
            delay(200);
            cleardevice();
            closegraph();
            user();
            cleardevice();
            closegraph();
            main1();
        }
        if(x>220&&x<410&&y>330&&y<370&&(button &1)==1)
        {
            delay(200);
            cleardevice();
            closegraph();
            newuser();
            home1();
```

```
            initmouse();
            showmouseptr();
        }
        if(x>90&&x<270&&y>390&&y<430&&(button &1)==1)
        {
            cleardevice();
            hidemouseptr();
            closegraph();
            graphics(0);
            setbkcolor(RED);
            setcolor(WHITE);
            settextstyle(2,HORIZ_DIR,8);
            settextjustify(1,1);
            outtextxy(getmaxx()/2,getmaxy()/2,"Thank you for using our
Software");
            delay(2000);
            closegraph();
            exit(0);
        }
        if(x>390&&x<570&&y>390&&y<430&&(button &1)==1)
        {
            delay(200);
            cleardevice();
            closegraph();
            info();
            home1();
            initmouse();
            showmouseptr();
        }
    }
}
struct XYScatterGraph
{
    int color;
    int points;
    double xpoints[1000];
    double ypoints[1000];
    double dataxmin;
    double dataxmax;
    double dataymin;
    double dataymax;
    double xlen;
    double ylen;
    XYScatterGraph() { }
    XYScatterGraph(int c, int numpoints);
    void FillPoints(double yarr[], double xarr[],int len);
    void Draw();
    void DrawGrid(int index = 1);
};
XYScatterGraph::XYScatterGraph(int c, int numpoints)
{
    if(numpoints > 1000)
        numpoints = 1000;
    points = numpoints;
    color = c;
    for(int i = 0; i < 1000; i++)
    {
        xpoints[i] = 0.0;
```

```
            ypoints[i] = 0.0;
        }
        dataxmin = 1000000;
        dataxmax = -1000000;
        dataymin = 1000000;
        dataymax = -1000000;
        xlen = 0;
        ylen = 0;
    }
    void XYScatterGraph::FillPoints(double yarr[],double xarr[], int len)
    {
        for(int i = 0; i < len; i++)
        {
            if(i > points)
                break;
            xpoints[i] = int(xarr[i]);
            ypoints[i] = yarr[i];
            if(dataymin > ypoints[i])
                dataymin = ypoints[i];
            if(dataymax < ypoints[i])
                dataymax = ypoints[i];
            if(dataxmin > xpoints[i])
                dataxmin = xpoints[i];
            if(dataxmax < xpoints[i])
                dataxmax = xpoints[i];
        }
        xlen = dataxmax - dataxmin;
        ylen = dataymax - dataymin;
        dataymax = dataymax + xlen / 75.0;
        dataymin = dataymin - xlen / 75.0;
        xlen = dataxmax - dataxmin;
        ylen = dataymax - dataymin;
    }
    void XYScatterGraph::DrawGrid(int index)
    {
        int xmax = getmaxx() - LTMARGIN * 2;
        int ymax = getmaxy() - TPMARGIN * 2;
        setcolor(WHITE);
        rectangle(LTMARGIN,TPMARGIN,LTMARGIN + xmax, TPMARGIN + ymax);
        setlinestyle(DOTTED_LINE, 1, 1);
        double xstart = dataxmin;
        double ystart = dataymax;
        for(int i = 0; i <= 10; i++)
        {
            double ypos = TPMARGIN +  (i / 10.0 * ymax);
            double xpos = LTMARGIN + (i / 10.0 * xmax);
            if( i != 10)
                line(LTMARGIN, ypos, LTMARGIN + xmax, ypos);
            line(xpos, TPMARGIN, xpos, TPMARGIN +  ymax);
            xstart = dataxmin +  (i) * xlen / 10;
            ystart = dataymax -  (i) * ylen / 10;
            char buf[128];
            if(i == 0)
                xstart = 0;
            sprintf(buf, "%3.1lf", xstart);
            if( (i % 2) == 0)
                outtextxy(xpos - 10, TPMARGIN + ymax + 10, buf);
            else
```

```
            outtextxy(xpos - 10, TPMARGIN + ymax + 30, buf);
        sprintf(buf, "%5.1lf", ystart);
        if(index == 1)
            outtextxy(LTMARGIN - 50, ypos - 5, buf);
        else
            outtextxy(LTMARGIN + xmax + 10 , ypos - 5, buf);
    }
}
void XYScatterGraph::Draw()
{
    int xmax = getmaxx() - LTMARGIN * 2;
    int ymax = getmaxy() - TPMARGIN * 2;
    int xold = 0;
    int yold = 0;
    setcolor(color);
    setlinestyle(SOLID_LINE, 1, 1);
    for(int j = 0; j < points; j++)
    {
        int xpos = LTMARGIN + (xpoints[j] - dataxmin) / xlen * xmax;
        int bottom = 480;
        int ypos = bottom - TPMARGIN - (ypoints[j] - dataymin) / ylen
* ymax;
        if(j > 0)
            line(xold, yold, xpos, ypos);
        xold = xpos;
        yold = ypos;
    }
}
class account
{
public:
    char acno[20];
    char name[20];
    int balance;
    account()
    {
        balance=0;
    }
    void putdata()
    {
        graphics(0);
        int maxx=getmaxx()/2;
        int maxy=getmaxy()/2;
        setbkcolor(RED);
        setcolor(WHITE);
        settextstyle(0,HORIZ_DIR,2);
        outtextxy(maxx-200,maxy-150,"ACCOUNT BALANCE");
        outtextxy(maxx-200,maxy-50,"User Name: ");
        outtextxy(maxx-42,maxy-50,name);
        outtextxy(maxx-200,maxy-10,"Account Number: ");
        outtextxy(maxx+42,maxy-10,acno);
        outtextxy(maxx-200,maxy+30,"Available Balance: ");
        char bal[8];
        sprintf(bal,"%d",balance);
        outtextxy(maxx+100,maxy+30,bal);
        outtextxy(maxx-200,maxy+75,"Press Enter to continue....");
        getch();
        cleardevice();
```

```
        closegraph();
    }
};
class shares
{
public:
    int nos;
    char compname[50];
    double fvalue;
    double cvalue;
    shares()
    {
        nos=0;
        fvalue=0.0;
        cvalue=0.0;
    }
    void putshares()
    {
        cout<<"Company name: ";
        puts(compname);
        cout<<"No.of Shares bought: "<<nos<<endl;
        cout<<"Share bought at: "<<setprecision(2)<<cvalue<<endl;
        cout<<"Face value of each share: "<<fvalue<<endl;
        rand_double(fvalue,cvalue);
        per(fvalue,cvalue);
        cout<<endl;
    }
};
class company
{
public:
    char cname[30];
    int ccode,nshares;
    double fvalue;
    double cvalue;
    void getdata();
    void putdata();
    int getcode()
    {
        return ccode;
    }
    char *getname()
    {
        return cname;
    }
};
void company::getdata()
{
    cout<<"Enter company name: ";
    gets(cname);
    cout<<"Enter the company code: ";
    cin>>ccode;
    cout<<"Enter the face value: ";
    cin>>fvalue;
    randomize();
    nshares=random(99);
}
void company::putdata()
```

```
{
    cout<<"Company name: ";
    puts(cname);
    cout<<"Code: "<<ccode<<endl;
    cout<<"Face value of each share: "<<fvalue<<endl;
    cout<<"Total shares available: "<<nshares<<endl;
    rand_double(fvalue,cvalue);
    per(fvalue,cvalue);
}
class newu
{
    char name[20];
    int age;
    char pan[20];
    char adr[100];
    char acno[20];
public:
    void get();
    char *getname()
    {
        return name;
    }
    char *getacno()
    {
        return acno;
    }
    void put1();
    void put()
    {
        graphics(0);
        int maxx=getmaxx()/2;
        int maxy=getmaxy()/2;
        setbkcolor(RED);
        setcolor(WHITE);
        settextstyle(0,HORIZ_DIR,1);
        outtextxy(maxx-200,maxy-150,"NAME:");
        outtextxy(maxx-25,maxy-150,name);
        outtextxy(maxx-200,maxy-120,"AGE:");
        char ch[10];
        sprintf(ch,"%d",age);
        outtextxy(maxx-25,maxy-120,ch);
        outtextxy(maxx-200,maxy-90,"PAN CARD NUMBER: ");
        outtextxy(maxx-25,maxy-90,pan);
        outtextxy(maxx-200,maxy-60,"ADDRESS: ");
        outtextxy(maxx-25,maxy-60,adr);
        outtextxy(maxx-200,maxy-30,"ACCOUNT NUMBER: ");
        outtextxy(maxx-25,maxy-30,acno);
        outtextxy(maxx-200,maxy+40,"Press Enter To Continue....");
        getch();
        cleardevice();
        closegraph();
    }
    void put2()
    {
        cout<<endl;
        cout<<"NAME:";
        puts(name);
        cout<<"AGE:"<<age<<endl;
```

```
        cout<<"PAN CARD NUMBER: ";
        puts(pan);
        cout<<"ADDRESS: ";
        puts(adr);
        cout<<"ACCOUNT NUMBER: ";
        puts(acno);
        cout<<endl;
    }
};
void newu::get()
{
name4:
    graphics(0);
    cleardevice();
    setbkcolor(BLUE);
    textcolor(WHITE);
    centerstring("Enter the name (less than 8 characters): ");
    gets(name);
    if(strlen(name)>7)
    {
        qe centerstring("Name should be less than 8 characters.");
        getch();
        memset(name,0,20);
        goto name4;
    }
    char *name2,*name3;
    name2=new char[50];
    name3=new char[50];
    strcpy(name3,name);
    strcat(name3,".dat");
    strcpy(name2,"C:/TC/BIN/StockE/");
    strcat(name2,name3);
    ifstream check1;
    check1.open(name2);
    delete[]name2;
    delete[]name3;
    if(!check1)
    {
age1:
        centerstring("Enter the age: ");
        cin>>age;
        if(cin.fail())
        {
            cin.clear();
            cin.ignore();
            qe centerstring("Enter a number not a character: ");
            getch();
            clrscr();
            cleardevice();
            goto age1;
        }
        if(age<18)
        {
            qe centerstring("You must be 18 and above to access this
platform.");
            getch();
            qe centerstring("ABORTING PROGRAM !!! (ERROR::User below
18)");
```

```
            delay(1500);
            cleardevice();
            closegraph();
            exit(0);
        }
        if(age>116)
        {
            centerstring("You are not Susannah Mushatt Jones !!");
            qe
            delay(2000);
            centerstring("Only mortals are allowed to use stock
exchange");
            qe
            delay(2500);
            centerstring("ABORTING PROGRAM !!! (ERROR::User is an
ASGUARDIAN)");
            delay(2500);
            cleardevice();
            closegraph();
            exit(0);
        }
pan1:
        centerstring("Enter pan card no. (10 characters): ");
        gets(pan);
        if(strlen(pan)!=10)
        {
            qe centerstring("Pan card number should be of 10
characters only.");
            getch();
            memset(pan,0,20);
            clrscr();
            cleardevice();
            goto pan1;
        }
        centerstring("Enter the address: ");
        gets(adr);
acno1:
        memset(acno,0,20);
        centerstring("Enter the bank account no. (9-18 digits) : ");
        gets(acno);
        for(int i=0; i<20; i++)
        {
            if(isalpha(acno[i])&&(!isdigit(acno[i])))
            {
                qe centerstring("Account number should only contain
digits.");
                getch();
                memset(acno,0,20);
                clrscr();
                cleardevice();
                goto acno1;
            }
        }
        if(strlen(acno)<9||strlen(acno)>18)
        {
            qe centerstring("Account number should be between 9 and 18
digits only.");
            getch();
```

```
            memset(acno,0,20);
            clrscr();
            cleardevice();
            goto acno1;
        }
    }
    else
    {
        centerstring("User already exists!");
        getch();
        memset(name,0,20);
        check1.close();
        cleardevice();
        closegraph();
        main1();
    }
}
void newu::put1()
{
lb:
    clrscr();
    char ch;
    textbackground(RED);
    textcolor(WHITE);
    centerstring("User menu ");
    qe qe
    centerstring("Enter 1 for displaying profile: ");
    qe qe
    centerstring("Enter 2 for editing your profile: ");
    qe qe
    centerstring("Enter 3 to manage stocks: ");
    qe qe
    centerstring("Enter 4 to view portfolio: ");
    qe qe
    centerstring("Enter 5 to view your account balance: ");
    qe qe
    centerstring("Enter 6 to logout: ");
    qe qe
    gotoxy(40,16);
    cin>>ch;
    if(cin.fail())
    {
        cin.clear();
        cin.ignore();
        cout<<"Enter a number not a character: "<<endl;
        delay(2000);
        goto lb;
    }
    switch(ch)
    {
    case '1':
        clrscr();
        put();
        goto lb;
    case '2':
        clrscr();
        newu m;
        cout<<"NAME: ";
```

```
        puts(name);
        strcpy(m.name,name);
        cout<<"AGE: "<<age;
        qe
        m.age=age;
        cout<<"PAN NO.: "<<pan;
        strcpy(m.pan,pan);
        qe
        centerstring("Enter the new address: ");
        qe
        centerstring("(.) to retain the current address");
        qe
        gotoxy(40,8);
        gets(m.adr);
        if(strcmp(m.adr,".")==0)
        {
            strcpy(m.adr,adr);
            goto acno2;
        }
acno2:
        memset(m.acno,0,20);
        centerstring("Enter the new bank account no. (9-18 digits) :
");
        qe
        centerstring("(.) to retain the current account no. ");
        qe
        gotoxy(40,11);
        gets(m.acno);
        if(strcmp(m.acno,".")==0)
        {
            strcpy(m.acno,acno);
        }
        else
        {
            for(int i=0; i<20; i++)
            {
                if(isalpha(m.acno[i])&&(!isdigit(m.acno[i])))
                {
                    centerstring("Account number should only contain
digits.");
                    qe
                    getch();
                    memset(m.acno,0,20);
                    clrscr();
                    goto acno2;
                }
            }
            if(strlen(m.acno)<9||strlen(m.acno)>18)
            {
                centerstring("Account number should be between 9 and
18 digits only.");
                qe
                getch();
                memset(m.acno,0,20);
                clrscr();
                goto acno2;
            }
        }
```

```
        strcpy(pan,m.pan);
        strcpy(adr,m.adr);
        strcpy(acno,m.acno);
        char *fname,*fname1;
        fname=new char[50];
        fname1=new char[50];
        strcpy(fname1,m.name);
        strcat(fname1,".dat");
        strcpy(fname,"C:/TC/BIN/StockE/");
        strcat(fname,fname1);
        ofstream file4,file5;
        file5.open(fname);
        remove(fname);
        file5.clear();
        file5.close();
        file4.open(fname,ios::app|ios::binary);
        file4.write((char*)&m,sizeof(m));
        file4.clear();
        file4.close();
        delete[]fname1;
        delete[]fname;
        clrscr();
        graphics(0);
        int x=getmaxx()/2;
        int y=getmaxy()/2;
        setbkcolor(RED);
        setcolor(WHITE);
        settextstyle(0,HORIZ_DIR,2);
        settextjustify(1,1);
        outtextxy(x,y,"Changes have been made to the profile");
        outtextxy(x,y+70,"Going to user menu...");
        delay(1250);
        cleardevice();
        closegraph();
        goto lb;
    case '3':
        stock(name);
        goto lb;
    case '4':
        portfolio(name);
        goto lb;
    case '6':
        graphics(0);
        int maxx=getmaxx()/2;
        int maxy=getmaxy()/2;
        setbkcolor(RED);
        setcolor(WHITE);
        settextstyle(0,HORIZ_DIR,2);
        settextjustify(1,1);
        outtextxy(maxx,maxy,"You have successfully logged out");
        outtextxy(maxx,maxy+50,name);
        delay(2000);
        cleardevice();
        break;
    case '5':
        clrscr();
        balance(name);
        goto lb;
```

```
    default:
        centerstring("Wrong choice");
        getch();
        clrscr();
        goto lb;
    }
}
void main1()
{
    int s=graphics(1);
    int maxx = getmaxx();
    int maxy = getmaxy();
    restrictmouseptr(1,1,maxx-1,maxy-1);
    s=initmouse();
    if ( s== 0 )
        cout<<"Mouse support not available.\n";
    else
        showmouseptr();
    Home d;
    d.getchoice_home();
}
void main()
{
    intro();
    main1();
}
void info()
{
    graphics(0);
    setbkcolor(MAGENTA);
    int maxx = getmaxx()/2;
    int maxy = getmaxy()/2;
    setcolor(WHITE);
    settextstyle(0,HORIZ_DIR,2);
    settextjustify(CENTER_TEXT,CENTER_TEXT);
    outtextxy(maxx,maxy,"Visit bseindia.com for more info.");
    getchar();
    main1();
}
void admin()
{
START:
    clrscr();
    graphics(0);
    setbkcolor(MAGENTA);
    int maxx = getmaxx()/2;
    int maxy = getmaxy()/2;
    setcolor(WHITE);
    settextstyle(0,HORIZ_DIR,2);
    outtextxy(maxx-200,maxy-150," Enter Password  : ");
    char pass[32],pass1[32];
    int i =0;
    int i1=-17;
    char a;
    for(i=0;;)
    {
        a=getch();
        if((a>='a'&&a<='z')||(a>='A'&&a<='Z')||(a>='0'&&a<='9'))
```

```
        {
            setcolor(WHITE);
            pass[i]=a;
            ++i;
            i1+=17;
            outtextxy(maxx-183+i1,maxy-120,"*");
        }
        if(a=='\b'&&i>=1)
        {
            cout<<"\b \b";
            setcolor(MAGENTA);
            outtextxy(maxx-183+i1,maxy-120,"*");
            --i;
            i1-=17;
        }
        if(a=='\r')
        {
            pass[i]='\0';
            break;
        }
    }
    if(i<=5)
    {
        settextjustify(1,1);
        outtextxy(maxx,maxy,"Minimum 6 digits needed.Enter Again");
        getch();
        goto START;
    }
    ifstream fin;
    fin.open("C:/TC/BIN/StockE/Password.dat");
    if(!fin)
    {
        cerr<<"error in opening of file"<<endl;
    }
    else
    {
        fin>>pass1;
        if(strcmp(pass,pass1)==0)
        {
            cleardevice();
            settextjustify(1,1);
            outtextxy(maxx,maxy,"Password accepted");
            delay(500);
            fin.close();
            adminfnt();
        }
        else
        {
            cleardevice();
            settextjustify(1,1);
            outtextxy(maxx,maxy,"Incorrect Password");
            delay(1000);
            fin.close();
            goto START;
        }
    }
    cleardevice();
    closegraph();
```

```
}
void adminfnt()
{
    clrscr();
    graphics(0);
    setbkcolor(MAGENTA);
    int maxx = getmaxx()/2;
    int maxy = getmaxy()/2;
    char ch,pass[32],pass1[32];
    time_t tim;
    time(&tim);
    settextstyle(0,HORIZ_DIR,3);
    settextjustify(1,1);
    outtextxy(maxx,maxy,"Welcome Administrator!! ");
    delay(750);
    cleardevice();
    settextjustify(1,1);
    settextstyle(0,HORIZ_DIR,1);
    textcolor(WHITE);
    cout<<setw(35)<<"Login Time : "<<ctime(&tim);
    outtextxy(maxx,maxy-180,"Enter 1 for List of Companies: ");
    outtextxy(maxx,maxy-160,"Enter 2 for Change password: ");
    outtextxy(maxx,maxy-140,"Enter 3 to manage User list: ");
    outtextxy(maxx,maxy-120,"Enter 4 to Log out: ");
    outtextxy(maxx,maxy-80,"Enter your choice: ");
    gotoxy(40,12);
    cin>>ch;
    switch(ch)
    {
    case '1':
        cleardevice();
        closegraph();
        textcolor(WHITE);
        textbackground(MAGENTA);
        lcom();
        gotoxy(40,12);
        break;
    case '4':
        cleardevice();
        settextstyle(0,HORIZ_DIR,2.5);
        settextjustify(1,1);
        outtextxy(maxx,maxy,"You have successfully logged out");
        delay(2000);
        cleardevice();
        closegraph();
        main1();
    case '2':
START1:
        clrscr();
        cleardevice();
        gotoxy(30,12);
        textcolor(WHITE);
        cout<<"Enter the old password: "<<endl;
        gotoxy(37,13);
        gets(pass);
        fstream f;
        f.open("C:/TC/BIN/StockE/Password.dat",ios::in);
        if(!f)
```

```
        {
            cerr<<"error in opening of file"<<endl;
        }
        else
        {
            f>>pass1;
            if(strcmp(pass,pass1)==0)
            {
                gotoxy(30,14);
                cout<<"Enter new password: "<<endl;
                int i = 0;
                char a,temp[32];
                for(i=0;;)
                {
                    a=getch();

if((a>='a'&&a<='z')||(a>='A'&&a<='Z')||(a>='0'&&a<='9'))
                    {
                        temp[i]=a;
                        ++i;
                        gotoxy(36+i,15);
                        cout<<"*";
                    }
                    if(a=='\b'&&i>=1)
                    {
                        cout<<"\b \b";
                        --i;
                    }
                    if(a=='\r')
                    {
                        temp[i]='\0';
                        break;
                    }
                }
                if(i<=5)
                {
                    settextjustify(1,1);
                    settextstyle(0,HORIZ_DIR,2.5);
                    outtextxy(maxx,maxy+100,"Minimum 6 digits
needed.Enter Again");
                    getch();
                    goto START1;
                }
                ofstream fout;
                fout.open("C:/TC/BIN/StockE/Password.dat");
                fout<<temp;
                fout.close();
                f.close();
                settextjustify(1,1);
                settextstyle(0,HORIZ_DIR,2.5);
                outtextxy(maxx,maxy+100,"Password changed");
                delay(1500);
                adminfnt();
            }
            else
            {
                settextjustify(1,1);
                settextstyle(0,HORIZ_DIR,2.5);
```

```
                outtextxy(maxx,maxy,"Incorrect Password ");
                delay(2000);
                adminfnt();
            }
        }
        break;
    case '3':
        cleardevice();
        closegraph();
        textcolor(WHITE);
        textbackground(MAGENTA);
        ucom();
        gotoxy(40,12);
        break;
    default:
        settextjustify(1,1);
        settextstyle(0,HORIZ_DIR,2.5);
        outtextxy(maxx,maxy,"Wrong choice");
        delay(1500);
        adminfnt();
        break;
    }
}
void newuser()
{
    clrscr();
    newu u;
    u.get();
    char *filename,*filename1;
    filename=new char[50];
    filename1=new char[50];
    strcpy(filename1,u.getname());
    strcat(filename1,".dat");
    strcpy(filename,"C:/TC/BIN/StockE/");
    strcat(filename,filename1);
    graphics(0);
    int maxx =getmaxx()/2;
    int maxy =getmaxy()/2;
    cleardevice();
    setbkcolor(CYAN);
    setcolor(WHITE);
    settextstyle(0,HORIZ_DIR,2.7);
    settextjustify(1,1);
    outtextxy(maxx,maxy,"You have successfully enlisted");
    outtextxy(maxx,maxy+35," as a new investor");
    delay(1250);
    char temp[100];
    strcpy(temp,filename);
    ofstream fout1,fout2;
    fout1.open(filename,ios::app|ios::binary);
    fout1.write((char*)&u,sizeof(u));
    fout1.close();
    getchar();
    fout2.open("C:/TC/BIN/StockE/Users.dat",ios::app);
    fout2<<temp<<endl;
    fout2.close();
    delete[]filename;
    delete[]filename1;
```

```
    account a;
    strcpy(a.name,u.getname());
    strcpy(a.acno,u.getacno());
    srand(time(NULL));
    int ranval=500+rand()%32000;
    a.balance=abs(ranval);
    char *aname,*aname1;
    aname=new char[50];
    aname1=new char[50];
    strcpy(aname1,a.name);
    strcat(aname1,"acco.dat");
    strcpy(aname,"C:/TC/BIN/StockE/");
    strcat(aname,aname1);
    ofstream fia;
    fia.open(aname,ios::ate|ios::binary);
    fia.write((char*)&a,sizeof(a));
    fia.close();
    delete[]aname;
    delete[]aname1;
    main1();
}
void user()
{
    clrscr();
    newu w;
    char *name1,*filename,*filename1;
    name1= new char[20];
    filename=new char[50];
    filename1=new char[50];
    graphics(0);
    int maxx=getmaxx()/2;
    int maxy=getmaxy()/2;
    settextstyle(0,HORIZ_DIR,3);
    settextjustify(1,1);
    setbkcolor(LIGHTRED);
    setcolor(WHITE);
    outtextxy(maxx,maxy,"Welcome to User Login !!");
    delay(1500);
    cleardevice();
    settextstyle(0,HORIZ_DIR,2.5);
    settextjustify(1,1);
    outtextxy(maxx,maxy,"Enter your name: ");
    char a,c[2];
    int i,i1=-17;
    for(i=0;;)
    {
        a=getch();
        sprintf(c,"%c",a);
        if((a>='a'&&a<='z')||(a>='A'&&a<='Z')||(a>='0'&&a<='9'))
        {
            setcolor(WHITE);
            name1[i]=a;
            ++i;
            i1+=17;
            settextjustify(1,1);
            outtextxy(maxx-68+i1,maxy+50,c);
        }
        if(a=='\b'&&i>=1)
```

```
            {
                cout<<"\b \b";
                setcolor(LIGHTRED);
                char ch1=219;
                sprintf(c,"%c",ch1);
                settextjustify(1,1);
                outtextxy(maxx-68+i1,maxy+50,c);
                --i;
                i1-=17;
            }
            if(a=='\r')
            {
                name1[i]='\0';
                break;
            }
        }
        strcpy(filename1,name1);
        strcat(filename1,".dat");
        strcpy(filename,"C:/TC/BIN/StockE/");
        strcat(filename,filename1);
        ifstream fin1;
        fin1.open(filename,ios::in|ios::binary);
        if(!fin1)
        {
            delete[]filename;
            delete[]filename1;
            delete[]name1;
            settextstyle(0,HORIZ_DIR,2.5);
            settextjustify(1,1);
            outtextxy(maxx,maxy+100,"User does not exist");
            getchar();
            cleardevice();
            main1();
        }
        else
        {
            delete[]filename;
            delete[]filename1;
            delete[]name1;
            fin1.read((char*)&w,sizeof(w));
            fin1.close();
            cleardevice();
            settextstyle(0,HORIZ_DIR,3);
            settextjustify(1,1);
            outtextxy(maxx-100,maxy,"Welcome");
            outtextxy(maxx+100,maxy,w.getname());
            outtextxy(maxx+120,maxy,"        !!");
            delay(1000);
            cleardevice();
            closegraph();
            textbackground(RED);
            textcolor(WHITE);
            w.put1();
        }
    }
    void lcom()
    {
    lb:
```

```
    clrscr();
    company com,obj;
    fstream file;
    int comcode;

file.open("C:/TC/BIN/StockE/company.dat",ios::in|ios::out|ios::ate|ios
::binary);
    int choice=1;
    clrscr();
    while(choice!=0)
    {
        centerstring("Enter 1 for adding a company");
        qe qe
        centerstring("Enter 2 for no. of companies");
        qe qe
        centerstring("Enter 3 for displaying the list");
        qe qe
        centerstring("Enter 4 to go to administrator menu");
        qe qe
        centerstring("Enter 5 to delete a company");
        qe qe
        centerstring("Enter your choice");
        qe qe
        gotoxy(40,14);
        cin>>choice;
        if(cin.fail())
        {
            cin.clear();
            cin.ignore();
            file.close();
            centerstring("Enter a number not a character: ");
            delay(2000);
            goto lb;
        }
        switch(choice)
        {
        case 1:
            clrscr();
            cout<<"Add a company"<<endl;
            com.getdata();
            file.write((char*)&com,sizeof(com));
            clrscr();
            break;
        case 2:
            clrscr();
            int filesize;
            filesize=file.tellg();
            int n=filesize/sizeof(com);
            cout<<"The no.of companies in the file are "<<n<<endl;
            getchar();
            clrscr();
            break;
        case 3:
            graphics(0);
            cleardevice();
            setbkcolor(MAGENTA);
            if(file.tellg()==0)
            {
```

```
                cout<<"No companies are present in the file"<<endl;
                cout<<"Press Enter to continue...."<<endl;
                getchar();
                clrscr();
                lcom();
                break;
            }
            else
            {
                file.seekg(0,ios::beg);
                cout<<"The current contents of the files are "<<endl;
                while(file.read((char*)&com,sizeof(com)))
                {
                    textcolor(WHITE);
                    com.putdata();
                    cout<<"Press Enter to continue...."<<endl;
                    getchar();
                }
                file.clear();
                cleardevice();
                closegraph();
                textcolor(WHITE);
                textbackground(MAGENTA);
            }
            clrscr();
            break;
        case 4:
            file.close();
            admin();
            break;
        case 5:
            clrscr();
            ofstream fout;
            fout.open("C:/TC/BIN/StockE/temp.dat",ios::binary);
            int ccode,check=0;
            cout<<"Enter company code whose record is to be
deleted"<<endl;
            cin>>ccode;
            file.seekg(0,ios::beg);
            while(file.read((char*)&com,sizeof(com)))
            {
                if(com.getcode()==ccode)
                {
                    check=1;
                }
                else
                {
                    fout.write((char*)&com,sizeof(com));
                }
            }
            file.close();
            fout.close();
            remove("C:/TC/BIN/StockE/company.dat");

rename("C:/TC/BIN/StockE/temp.dat","C:/TC/BIN/StockE/company.dat");

file.open("C:/TC/BIN/StockE/company.dat",ios::in|ios::out|ios::ate|ios
::binary);
```

```
            if(check==0)
            {
                cout<<"Company code not found"<<endl;
            }
            else
            {
                cout<<"Company successfully deleted"<<endl;
            }
            getch();
            clrscr();
            break;
        default:
            gotoxy(35,16);
            cout<<"Wrong choice";
            file.close();
            getch();
            clrscr();
            break;
        }
    }
    getch();
}
void ucom()
{
lb0:
    clrscr();
    newu q;
    char loc[100],loc2[100];
    int choice=1;
    while(choice!=0)
    {
        clrscr();
        centerstring("Enter 1 for no. of users");
        qe qe
        centerstring("Enter 2 for displaying the users");
        qe qe
        centerstring("Enter 3 for deleting users");
        qe qe
        centerstring("Enter 4 to go to administrator menu");
        qe qe
        centerstring("Enter your choice");
        qe qe
        gotoxy(40,12);
        cin>>choice;
        if(cin.fail())
        {
            cin.clear();
            cin.ignore();
            centerstring("Enter a number not a character: ");
            delay(2000);
            goto lb0;
        }
        switch(choice)
        {
        case 1:
            clrscr();
            int count=0;
            fstream file;
```

```
        file.open("C:/TC/BIN/StockE/Users.dat",ios::in);
        if(!file)
        {
            cerr<<"File does not exist"<<endl;
            delay(1000);
            clrscr();
            file.close();
            goto lb0;
        }
        while(!file.eof())
        {
            file.getline(loc,100);
            if(loc[0]=='C')
                count++;
        }
        file.close();
        cout<<"The no.of users in the file are "<<count<<endl;
        getchar();
        clrscr();
        break;
    case 2:
        clrscr();
        fstream file1,file2;
        file1.open("C:/TC/BIN/StockE/Users.dat",ios::in);
        if(!file1)
        {
            cerr<<"File not found"<<endl;
            delay(1250);
            clrscr();
            goto lb0;
        }
        int p=0;
        while(!file1.eof())
        {
            file1.getline(loc2,100);
            if(loc2[0]=='C')
                p++;
        }
        file1.close();
        if(p==0)
        {
            cout<<"No users present in the database"<<endl;
            delay(1250);
            clrscr();
            goto lb0;
        }
        else
        {
            file2.open("C:/TC/BIN/StockE/Users.dat",ios::in);
            cout<<"The current contents of the files are "<<endl;
            getchar();
            fstream fin;
            for(int i=0; i<50; i++)
            {
                file2.getline(loc2,100);
                fin.open(loc2,ios::in|ios::binary);
                if(!fin)
                    continue;
```

```
                fin.read((char*)&q,sizeof(q));
                q.put2();
                cout<<"Press Enter to continue...."<<endl;
                getchar();
                fin.clear();
                fin.close();
            }
            file2.clear();
            file2.close();
        }
        clrscr();
        break;
    case 3:
        clrscr();
        char *user,*filename,*filename1;
        user= new char[20];
        filename=new char[50];
        filename1=new char[50];
        cout<<"Enter the user name to be deleted: "<<endl;
        gets(user);
        strcpy(filename1,user);
        strcat(filename1,".dat");
        strcpy(filename,"C:/TC/BIN/StockE/");
        strcat(filename,filename1);
        fstream fins;
        fins.open(filename,ios::in);
        if(!fins)
        {
            cout<<"User not found"<<endl;
            delay(1500);
            clrscr();
            goto lb0;
        }
        else
        {
            cout<<"User successfully deleted"<<endl;
            fins.close();
            remove(filename);
            char *pname,*pname1,*aname,*aname1;
            pname=new char[50];
            pname1=new char[50];
            aname=new char[50];
            aname1=new char[50];
            strcpy(pname1,user);
            strcpy(aname1,user);
            strcat(pname1,"share.dat");
            strcat(aname1,"acco.dat");
            strcpy(pname,"C:/TC/BIN/StockE/");
            strcpy(aname,pname);
            strcat(pname,pname1);
            strcat(aname,aname1);
            fstream share,acco;
            share.open(pname,ios::binary);
            remove(pname);
            acco.open(aname,ios::binary);
            remove(aname);
            delay(1500);
            share.close();
```

```
                acco.close();
                clrscr();
                delete[]pname;
                delete[]pname1;
                delete[]aname;
                delete[]aname1;
            }
            fstream file3;
            file3.open("C:/TC/BIN/StockE/Users.dat",ios::in);
            ofstream fout;
            fout.open("C:/TC/BIN/StockE/temp.dat",ios::out);
            while(file3.eof()==0)
            {
                file3.getline(loc2,100);
                if(strcmp(loc2,filename)!=0)
                {
                    fout<<loc2<<endl;
                }
            }
            file3.close();
            fout.close();
            remove("C:/TC/BIN/StockE/Users.dat");

rename("C:/TC/BIN/StockE/temp.dat","C:/TC/BIN/StockE/Users.dat");
            delete[]filename;
            delete[]user;
            delete[]filename1;
            break;
        case 4:
            admin();
            break;
        default:
            centerstring("Wrong choice");
            getch();
            clrscr();
            gotoxy(40,12);
            break;
        }
    }
}
void stock(char name[20])
{
stock:
    clrscr();
    char d;
    centerstring("Enter 1 to buy shares");
    qe qe
    centerstring("Enter 2 to sell shares");
    qe qe
    centerstring("Enter 3 to go back to user menu");
    qe qe
    gotoxy(40,9);
    cin>>d;
    switch(d)
    {
    case '1':
        clrscr();
        buy(name);
```

```
        break;
    case '2':
        clrscr();
        sell(name);
        break;
    case '3':
        clrscr();
        break;
    default:
        qe centerstring("Wrong choice ");
        getch();
        clrscr();
        goto stock;
    }
}
void per(double fvalue,double cvalue)
{
    if(cvalue<fvalue)
    {
        double p;
        p=double(100.0-(cvalue/fvalue)*100.0);
        cout<<"Current value of each share:
"<<setprecision(2)<<cvalue;
        qe
        cout<<"Decrease: "<<setprecision(2)<<double(cvalue-
fvalue)<<endl;
        cout<<"Percentage decrease: "<<setprecision(2)<<"-
"<<double(p)<<"%"<<endl;
    }
    else if(cvalue>fvalue)
    {
        double p;
        p=double(100.0-(fvalue/cvalue)*100.0);
        cout<<"Current value of each share:
"<<setprecision(2)<<cvalue;
        qe
        cout<<"Increase: "<<setprecision(2)<<"+"<<double(cvalue-
fvalue)<<endl;
        cout<<"Percentage increase:
"<<setprecision(2)<<"+"<<double(p)<<"%"<<endl;
    }
    else
    {
        cout<<"No change"<<endl;
    }
}
void rand_double(double f,double&c)
{
    int x,y;
    randomize();
    x=1+random(100);
    if(f>x)
    {
        randomize();
        y=1+random(100);
    }
    else if(f<x)
    {
```

```
            randomize();
            y=1+random(x);
        }
        else
        {
            y=x;
        }
        srand(time(NULL));
        double ran_val=1.0*rand()/(RAND_MAX/2)+rand()%y;
        if(x<50)
        {
            c=double(f+ran_val);
        }
        else
        {
            c=abs(double(f-ran_val));
        }
    }
    void portfolio(char n[20])
    {
        graphics(0);
        cleardevice();
        setbkcolor(RED);
        textcolor(WHITE);
        shares s2;
        char *pname,*pname1;
        pname=new char[50];
        pname1=new char[50];
        strcpy(pname1,n);
        strcat(pname1,"share.dat");
        strcpy(pname,"C:/TC/BIN/StockE/");
        strcat(pname,pname1);
        ifstream inshare;
        inshare.open(pname,ios::ate|ios::binary);
        if(!inshare)
        {
            cout<<"Press Enter to continue...."<<endl;
            getchar();
            goto f;
        }
        else if(inshare.tellg()==0)
        {
            graphics(0);
            settextstyle(0,HORIZ_DIR,2.5);
            int maxx=getmaxx()/2;
            int maxy=getmaxy()/2;
            setbkcolor(RED);
            setcolor(WHITE);
            settextjustify(1,1);
            outtextxy(maxx,maxy,"No shares bought yet");
            getch();
            goto f;
        }
        else
        {
            settextstyle(0,HORIZ_DIR,2);
            settextjustify(1,1);
            outtextxy(getmaxx()/2,(getmaxx()/2)-300,"PORTFOLIO");
```

```
        qe qe qe qe
        inshare.seekg(0,ios::beg);
        while(inshare.read((char*)&s2,sizeof(s2)))
        {
            textcolor(WHITE);
            s2.putshares();
            cout<<endl;
            cout<<"Press Enter to continue...."<<endl;
            getchar();
        }
    }
f:
    cleardevice();
    closegraph();
    inshare.close();
    clrscr();
    textcolor(WHITE);
    textbackground(RED);
}
void balance(char name[20])
{
    clrscr();
    account a1;
    char *bname,*bname1;
    bname=new char[50];
    bname1=new char[50];
    strcpy(bname1,name);
    strcat(bname1,"acco.dat");
    strcpy(bname,"C:/TC/BIN/StockE/");
    strcat(bname,bname1);
    ifstream acc;
    acc.open(bname,ios::binary);
    if(!acc)
    {
        cout<<"Press Enter to continue...."<<endl;
        getchar();
        clrscr();
        goto balance;
    }
    acc.seekg(0,ios::beg);
    acc.read((char*)&a1,sizeof(a1));
    a1.putdata();
balance:
    acc.close();
    delete[]bname;
    delete[]bname1;
}
void buy(char name[20])
{
buy:
    graphics(0);
    setbkcolor(RED);
    setcolor(WHITE);
    company e;
    int code;
    char a;
    fstream fis;
```

```
fis.open("C:/TC/BIN/StockE/company.dat",ios::in|ios::out|ios::ate|ios:
:binary);
    centerstring("Enter 1 to enter the company code: ");
    qe qe
    centerstring("Enter 2 to display the list of companies: ");
    qe qe
    centerstring("Enter 3 to go back to user menu: ");
    qe qe
    gotoxy(40,9);
    cin>>a;
    switch(a)
    {
    case '1':
        restorecrtmode();
        textbackground(RED);
        textcolor(WHITE);
        int check=1;
        clrscr();
        centerstring("Enter the company code");
        qe
        gotoxy(40,3);
        cin>>code;
        fis.seekg(0,ios::beg);
        while(fis.read((char*)&e,sizeof(e)))
        {
            if(e.getcode()==code)
            {
                check=0;
                break;
            }
        }
        if(check==1)
        {
            qe centerstring("Company with this code does not exist");
            qe
            getchar();
            clrscr();
            fis.close();
            goto buy;
        }
        else
        {
            e.putdata();
            qe
            cout<<"Press enter to continue....";
            qe
            getch();
            check=0;
gb:
            char ch;
            clrscr();
            centerstring("Enter 1 to view the graph of this company");
            qe
            centerstring("Enter 2 to continue buying");
            qe
            gotoxy(40,4);
            cin>>ch;
```

```
switch(ch)
{
case '1':
    graphics(0);
    char code[4],fval[10],cval[10];
    outtextxy(10,20,"Company Name:");
    outtextxy(130,20,e.cname);
    outtextxy(10,40,"Security Code:");
    sprintf(code,"%d",e.ccode);
    outtextxy(130,40,code);
    memset(code,0,4);
    outtextxy(10,60,"Face Value:");
    sprintf(fval,"%.2f",e.fvalue);
    outtextxy(130,60,fval);
    memset(fval,0,10);
    outtextxy(10,80,"Current Value:");
    sprintf(cval,"%.2f",e.cvalue);
    outtextxy(130,80,cval);
    memset(cval,0,10);
    outtextxy(450,20,"x-axis: Time(Minutes)");
    outtextxy(450,40,"y-axis: Current Value");
    int color=getcolor();
    if(e.cvalue>e.fvalue)
    {
        upg();
        setcolor(color);
    }
    else if(e.cvalue<e.fvalue)
    {
        downg();
        setcolor(color);
    }
    double yarr[100],xarr[100];
    int t=0,x=0;
    for(int i=0; i<99; i++)
    {
        if(t%10==0)
        {
            t++;
            yarr[i]=e.cvalue+random(100);
        }
        else if(t%5==0)
        {
            t++;
            yarr[i]=abs(e.cvalue-random(100));
        }
        else
        {
            yarr[i]=e.fvalue;
            t++;
        }
        xarr[i]=x;
        x=x+1;
    }
    yarr[i]=e.cvalue;
    xarr[i]=100.0;
    setbkcolor(BLACK);
    int colour[]= {1,2,3,4,5,7,20,57,58,59,60,61,62,63};
```

```
                randomize();
                i=random(14);
                XYScatterGraph myGraph1(colour[i],100);
                myGraph1.FillPoints(yarr,xarr,100);
                myGraph1.DrawGrid(1);
                myGraph1.Draw();
                getch();
                closegraph();
                restorecrtmode();
                goto gb;
            case '2':
nss:
                clrscr();
                char *sname,*sname1;
                sname=new char[50];
                sname1=new char[50];
                strcpy(sname1,name);
                strcat(sname1,"share.dat");
                strcpy(sname,"C:/TC/BIN/StockE/");
                strcat(sname,sname1);
                ofstream fshare;
                fshare.open(sname,ios::ate|ios::binary);
                delete[]sname1;
                delete[]sname;
                shares s1;
                int ns;
                centerstring("Enter the no.of shares you want to
buy");
                qe
                gotoxy(40,3);
                cin>>ns;
                if(cin.fail())
                {
                    cin.clear();
                    cin.ignore();
                    qe centerstring("Enter a number not a character:
");
                    getch();
                    clrscr();
                    fshare.close();
                    goto nss;
                }
                if(e.nshares==0)
                {
                    qe centerstring("Sorry this Company has no shares
available at this time!!");
                    qe
                    getch();
                    qe centerstring("Contact admin to delist the
company");
                    qe
                    getch();
                    clrscr();
                    fshare.close();
                    fis.close();
                    break;
                }
                else if(ns>e.nshares)
```

```
                {
                    qe centerstring("Number of shares exceeds the
available lot.");
                    getchar();
                    clrscr();
                    fshare.close();
                    goto nss;
                }
                else if(ns<=0)
                {
                    qe centerstring("Number of shares cannot be
negative or 0.");
                    getchar();
                    clrscr();
                    fshare.close();
                    fis.close();
                    break;
                }
                else if(e.nshares<0)
                {
                    qe centerstring("Available shares cannot be
negative");
                    getch();
                    qe centerstring("Contact Admin to resolve this
issue");
                    getch();
                    clrscr();
                    fshare.close();
                    fis.close();
                    break;
                }
                else
                {
                    account a2;
                    char *bname,*bname1;
                    bname=new char[50];
                    bname1=new char[50];
                    strcpy(bname1,name);
                    strcat(bname1,"acco.dat");
                    strcpy(bname,"C:/TC/BIN/StockE/");
                    strcat(bname,bname1);
                    ifstream acc;
                    acc.open(bname,ios::binary);
                    acc.read((char*)&a2,sizeof(a2));
                    remove(bname);
                    acc.close();
                    a2.putdata();
                    a2.balance-=int(ns*e.cvalue);
                    if(a2.balance<500)
                    {
                        clrscr();
                        graphics(0);
                        setbkcolor(RED);
                        setcolor(WHITE);
                        cout<<"Transaction could not be completed.
(ERROR::Not enough cash)"<<endl;
                        getch();
```

```
                        cout<<"Add the required cash to complete the
transaction"<<endl;
                        getch();
                        fshare.close();
                        fis.close();
                        textbackground(RED);
                        textcolor(WHITE);
                        break;
                    }
                    graphics(0);
                    int maxx=getmaxx()/2;
                    int maxy=getmaxy()/2;
                    setbkcolor(RED);
                    setcolor(WHITE);
                    settextstyle(0,HORIZ_DIR,2);
                    outtextxy(maxx-200,maxy-150,"UPDATED BALANCE");
                    outtextxy(maxx-200,maxy-50,"User Name: ");
                    outtextxy(maxx-42,maxy-50,a2.name);
                    outtextxy(maxx-200,maxy-10,"Account Number: ");
                    outtextxy(maxx+42,maxy-10,a2.acno);
                    outtextxy(maxx-200,maxy+30,"Available Balance: ");
                    char bal[8];
                    sprintf(bal,"%d",a2.balance);
                    outtextxy(maxx+100,maxy+30,bal);
                    outtextxy(maxx-200,maxy+75,"Press Enter to
continue....");
                    getch();
                    cleardevice();
                    settextjustify(1,1);
                    settextstyle(0,HORIZ_DIR,3);
                    outtextxy(maxx,maxy,"Purchase Successful!!");
                    delay(1000);
                    cleardevice();
                    closegraph();
                    restorecrtmode();
                    ofstream acc1;
                    acc1.open(bname,ios::binary);
                    acc1.write((char*)&a2,sizeof(a2));
                    acc1.close();
                    delete[]bname;
                    delete[]bname1;
                    e.nshares-=ns;
                    s1.nos=ns;
                    s1.cvalue=e.cvalue;
                    strcpy(s1.compname,e.cname);
                    s1.fvalue=e.fvalue;
                    fshare.write((char*)&s1,sizeof(s1));
                    company y;
                    ofstream fit;
                    fit.open("C:/TC/BIN/StockE/temp.dat",ios::binary);
                    fis.seekg(0,ios::beg);
                    while(fis.read((char*)&y,sizeof(y)))
                    {
                        if(y.getcode()!=e.getcode())
                        {
                            fit.write((char*)&y,sizeof(y));
                        }
                    }
```

```
                    fit.write((char*)&e,sizeof(e));
                    fis.close();
                    fit.close();
                    fshare.close();
                    remove("C:/TC/BIN/StockE/company.dat");
rename("C:/TC/BIN/StockE/temp.dat","C:/TC/BIN/StockE/company.dat");
                    buycheck(name,s1.compname);
                }
                break;
            default:
                centerstring("Wrong Choice");
                getch();
                goto gb;
            }
            fis.close();
            goto buy;
        }
    case '2':
        graphics(0);
        cleardevice();
        setbkcolor(RED);
        fis.seekg(0,ios::beg);
        while(fis.read((char*)&e,sizeof(e)))
        {
            textcolor(WHITE);
            e.putdata();
            cout<<"Press Enter to continue...."<<endl;
            getchar();
        }
        fis.clear();
        fis.close();
        cleardevice();
        closegraph();
        clrscr();
        goto buy;
    case '3':
        fis.close();
        cleardevice();
        closegraph();
        clrscr();
        break;
    default:
        centerstring("Wrong choice ");
        qe
        getch();
        fis.close();
        goto buy;
    }
}
void sell(char name[20])
{
sell:
    clrscr();
    fstream fib;
    char t;
    company h;
```

```
fib.open("C:/TC/BIN/StockE/company.dat",ios::in|ios::out|ios::ate|ios:
:binary);
    centerstring("Enter 1 to enter the name of the company");
    qe qe
    centerstring("Enter 2 to display your portfolio");
    qe qe
    centerstring("Enter 3 to go back to user menu");
    qe qe
    gotoxy(40,7);
    cin>>t;
    switch(t)
    {
    case '1':
        char n[20];
        int scheck=1,scheck1=1;
        clrscr();
        centerstring("Enter the name of the company: ");
        qe
        gotoxy(40,3);
        gets(n);
        fib.seekg(0,ios::beg);
        while(fib.read((char*)&h,sizeof(h)))
        {
            if(strcmp(h.getname(),n)==0)
            {
                scheck=0;
                break;
            }
        }
        if(scheck==1)
        {
            centerstring("This company name does not exist");
            qe
            getchar();
            clrscr();
            fib.close();
            goto sell;
        }
        else
        {
            shares s3;
            char *pname,*pname1;
            pname=new char[50];
            pname1=new char[50];
            strcpy(pname1,name);
            strcat(pname1,"share.dat");
            strcpy(pname,"C:/TC/BIN/StockE/");
            strcat(pname,pname1);
            fstream inshare1;

inshare1.open(pname,ios::in|ios::out|ios::ate|ios::binary);
            delete[]pname1;
            inshare1.seekg(0,ios::beg);
            while(inshare1.read((char*)&s3,sizeof(s3)))
            {
                if(strcmp(s3.compname,n)==0)
                {
```

```
                    scheck1=0;
                    break;
                }
            }
            if(scheck1==1)
            {
                centerstring("No transaction has taken place with this
company");
                qe
                getch();
                clrscr();
                fib.close();
                inshare1.close();
                delete[]pname;
                goto sell;
            }
            else
            {
                s3.putshares();
                getch();
                clrscr();
                int ns;
                centerstring("Enter the no.of shares you want to sell:
");
                qe
                gotoxy(40,3);
                cin>>ns;
                if(s3.nos>=ns&&ns>0)
                {
                    account a1;
                    char *bname,*bname1;
                    bname=new char[50];
                    bname1=new char[50];
                    strcpy(bname1,name);
                    strcat(bname1,"acco.dat");
                    strcpy(bname,"C:/TC/BIN/StockE/");
                    strcat(bname,bname1);
                    fstream acc;
                    acc.open(bname,ios::in|ios::binary);
                    acc.seekg(0,ios::beg);
                    acc.read((char*)&a1,sizeof(a1));
                    remove(bname);
                    acc.close();
                    a1.putdata();
                    a1.balance+=int(ns*h.cvalue);
                    graphics(0);
                    int maxx=getmaxx()/2;
                    int maxy=getmaxy()/2;
                    setbkcolor(RED);
                    setcolor(WHITE);
                    settextstyle(0,HORIZ_DIR,2);
                    outtextxy(maxx-200,maxy-150,"UPDATED BALANCE");
                    outtextxy(maxx-200,maxy-50,"User Name: ");
                    outtextxy(maxx-42,maxy-50,a1.name);
                    outtextxy(maxx-200,maxy-10,"Account Number: ");
                    outtextxy(maxx+42,maxy-10,a1.acno);
                    outtextxy(maxx-200,maxy+30,"Available Balance: ");
                    char bal[8];
```

```
                    sprintf(bal,"%d",a1.balance);
                    outtextxy(maxx+100,maxy+30,bal);
                    outtextxy(maxx-200,maxy+75,"Press Enter to
continue....");
                    getch();
                    cleardevice();
                    settextjustify(1,1);
                    settextstyle(0,HORIZ_DIR,3);
                    outtextxy(maxx,maxy,"Transaction Successful!!");
                    delay(1000);
                    cleardevice();
                    closegraph();
                    restorecrtmode();
                    ofstream acc1;
                    acc1.open(bname,ios::binary);
                    acc1.write((char*)&a1,sizeof(a1));
                    acc1.close();
                    delete[]bname;
                    delete[]bname1;
                    s3.nos-=ns;
                    shares s4;
                    fstream fsell;
                    ofstream ftemp,ftemp1;

fsell.open(pname,ios::in|ios::out|ios::ate|ios::binary);

ftemp.open("C:/TC/BIN/StockE/temp.dat",ios::binary);
                    fsell.seekg(0,ios::beg);
                    while(fsell.read((char*)&s4,sizeof(s4)))
                    {
                        if(strcmp(s4.compname,s3.compname)!=0)
                            ftemp.write((char*)&s4,sizeof(s4));
                    }
                    ftemp.write((char*)&s3,sizeof(s3));
                    fsell.close();
                    ftemp.close();
                    remove(pname);
                    rename("C:/TC/BIN/StockE/temp.dat",pname);
                    h.nshares+=ns;
                    company z;

ftemp1.open("C:/TC/BIN/StockE/temp.dat",ios::binary);
                    fib.seekg(0,ios::beg);
                    while(fib.read((char*)&z,sizeof(z)))
                    {
                        if(z.getcode()!=h.getcode())
                            ftemp1.write((char*)&z,sizeof(z));
                    }
                    ftemp1.write((char*)&h,sizeof(h));
                    fib.close();
                    ftemp1.close();
                    remove("C:/TC/BIN/StockE/company.dat");

rename("C:/TC/BIN/StockE/temp.dat","C:/TC/BIN/StockE/company.dat");
                    clrscr();
                    sellcheck(pname);
                    delete[]pname;
                }
```

```
                else if(ns<=0)
                {
                    centerstring("Number of shares cannot be negative
or 0.");
                    qe
                    getch();
                    fib.close();
                    delete[]pname;
                    break;
                }
                inshare1.close();
                goto sell;
            }
        }
    case '2':
        portfolio(name);
        goto sell;
    case '3':
        fib.close();
        clrscr();
        break;
    default:
        qe centerstring("Wrong choice ");
        getch();
        fib.close();
        goto sell;
    }
}
void buycheck(char name[20],char cname[20])
{
    char *dname,*dname1;
    dname=new char[50];
    dname1=new char[50];
    strcpy(dname1,name);
    strcat(dname1,"share.dat");
    strcpy(dname,"C:/TC/BIN/StockE/");
    strcat(dname,dname1);
    fstream inshare2;
    int d=0,unos=0;
    double cur;
    shares s5,s6;
    inshare2.open(dname,ios::in|ios::out|ios::ate|ios::binary);
    delete[]dname1;
    inshare2.seekg(0,ios::beg);
    while(inshare2.read((char*)&s5,sizeof(s5)))
    {
        if(strcmp(s5.compname,cname)==0)
        {
            d++;
            if(d==1)
            {
                s6=s5;
            }
            if(d==2)
            {
                unos=s5.nos;
                cur=s5.cvalue;
            }
```

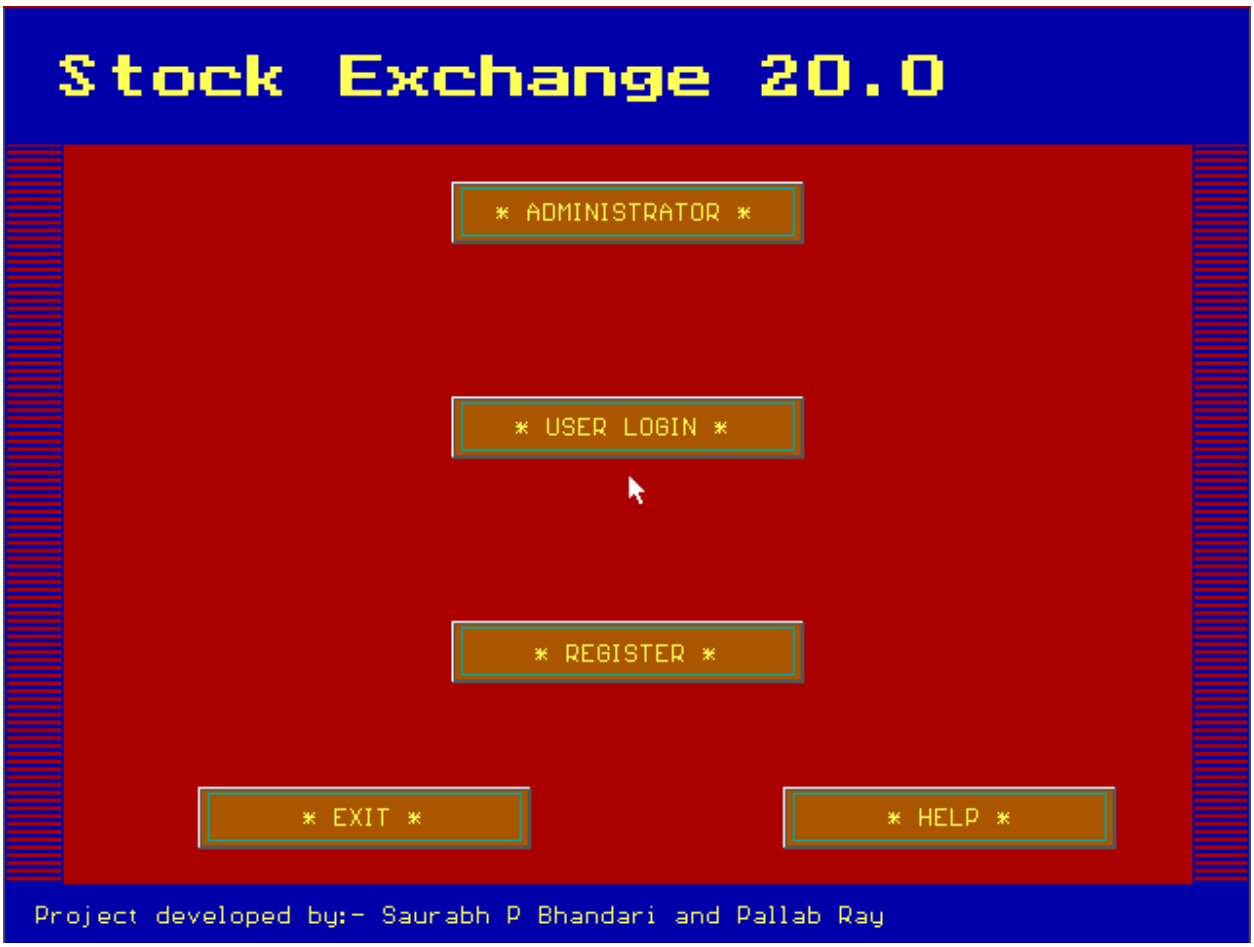
```
        }
    }
    if(d==2)
    {
        s6.nos+=unos;
        s6.cvalue=cur;
        shares s7;
        inshare2.close();
        fstream inshare3;
        inshare3.open(dname,ios::in|ios::out|ios::ate|ios::binary);
        ofstream ftemp;
        ftemp.open("C:/TC/BIN/StockE/temp.dat",ios::binary);
        inshare3.seekg(0,ios::beg);
        while(inshare3.read((char*)&s7,sizeof(s7)))
        {
            if(strcmp(s7.compname,s6.compname)!=0)
                ftemp.write((char*)&s7,sizeof(s7));
        }
        ftemp.write((char*)&s6,sizeof(s6));
        inshare3.close();
        ftemp.close();
        remove(dname);
        rename("C:/TC/BIN/StockE/temp.dat",dname);
        delete[]dname;
    }
    else
    {
        delete[]dname;
        inshare2.close();
    }
}
void sellcheck(char*pname)
{
    cout<<"Checking for existing companies having 0 zero shares in
your portfolio"<<endl;
    for(int i=0; i<7; i++)
    {
        cout<<".";
        delay(250);
    }
    shares s8;
    int check=1;
    fstream inshare4;
    inshare4.open(pname,ios::in|ios::out|ios::ate|ios::binary);
    ofstream ftemp;
    ftemp.open("C:/TC/BIN/StockE/temp.dat",ios::binary);
    inshare4.seekg(0,ios::beg);
    while(inshare4.read((char*)&s8,sizeof(s8)))
    {
        if(s8.nos==0)
        {
            check=0;
        }
        else
        {
            ftemp.write((char*)&s8,sizeof(s8));
        }
    }
```

```
    inshare4.close();
    ftemp.close();
    remove(pname);
    rename("C:/TC/BIN/StockE/temp.dat",pname);
    if(check==1)
    {
        cout<<endl<<"Your portfolio is alright."<<endl;
        getch();
    }
    else
    {
        cout<<endl<<"The company you sold now has 0 shares in your
portfolio."<<endl;
        getch();
        cout<<"Portfolio has been successfully repaired!!"<<endl;
        getch();
    }
}
void upg()
{
    char v[2];
    char t=30;
    sprintf(v,"%c",t);
    setcolor(GREEN);
    outtextxy(180,80,v);
    memset(v,0,2);
}
void downg()
{
    char p[2];
    char t=31;
    sprintf(p,"%c",t);
    setcolor(RED);
    outtextxy(180,80,p);
    memset(p,0,2);
}
void centerstring(char*s)
{
    int l=strlen(s);
    int pos=(int)((80-l)/2);
    for(int i=0; i<pos; i++)
        cout<<" ";
    cout<<s;
}
void intro()
{
    graphics(0);
    setbkcolor(GREEN);
    int maxx = getmaxx()/2;
    int maxy = getmaxy()/2;
    setcolor(WHITE);
    settextstyle(0,HORIZ_DIR,1);
    outtextxy(maxx-293,maxy-180,"Never invested in stocks and shares
before?");
    outtextxy(maxx-293,maxy-160,"Want to start now? Try our custom
Stock Exchange.");
    outtextxy(maxx-293,maxy-140,"Buy and sell shares and try to make a
profit.");
```

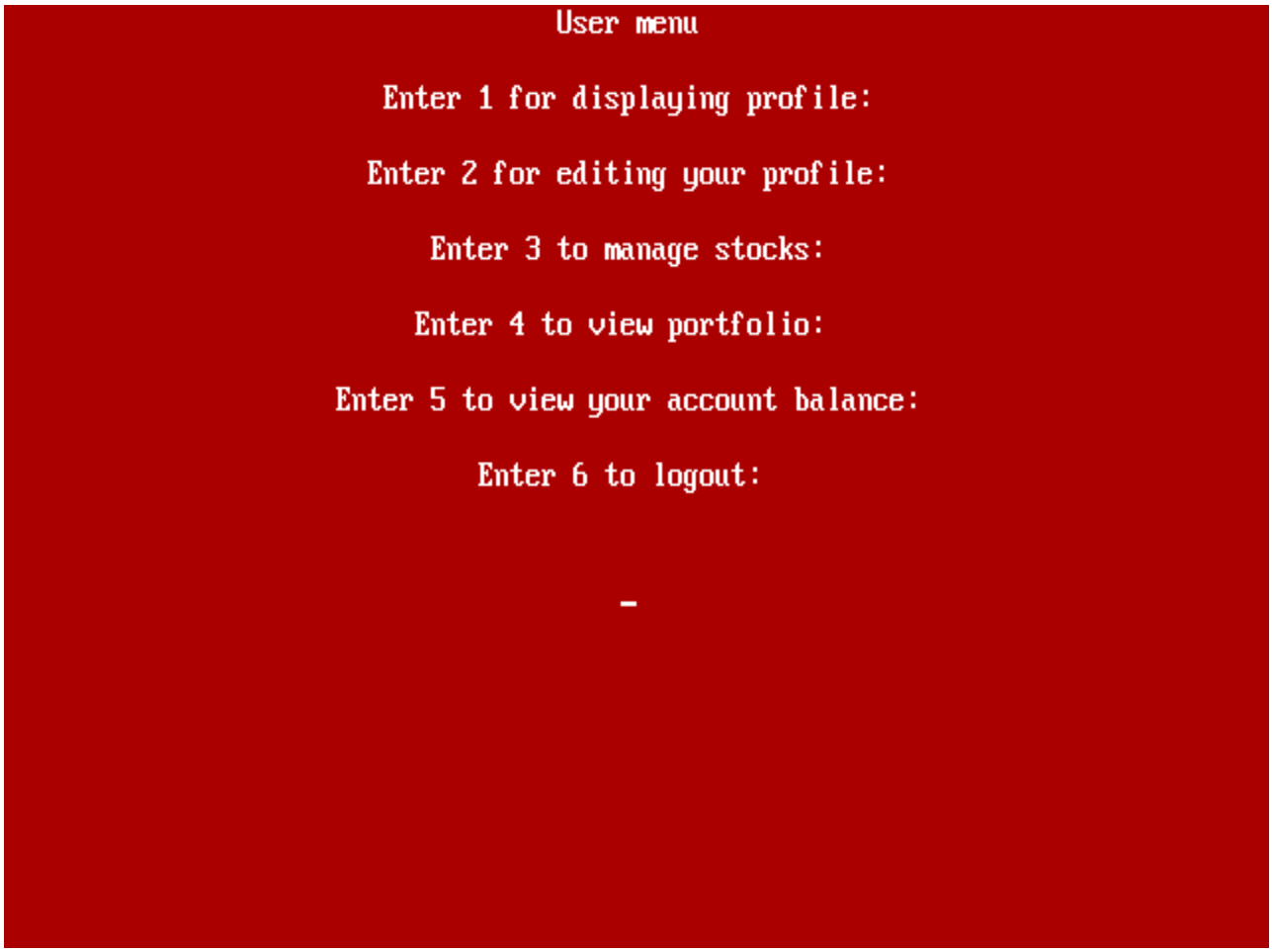
```
    outtextxy(maxx-293,maxy-120,"Register yourself as an user and keep
yourself updated on the increase ");
    outtextxy(maxx-293,maxy-100,"decrease of stocks you own. You can
even view the graph of the company ");
    outtextxy(maxx-293,maxy-80,"you want to buy shares from. Check
your account balance and find out how ");
    outtextxy(maxx-293,maxy-60,"much credit you have left. Don't be
worried if you incur a loss.");
    outtextxy(maxx-293,maxy-40,"You can always start again.");
    settextjustify(1,1);
    outtextxy(maxx,maxy,"Press enter to continue");
    getch();
    cleardevice();
    closegraph();
}
int graphics(int i)
{
    int status, gd=DETECT, gm;
    initgraph(&gd,&gm,"C:\\TC\\BGI");
    if(i==1)
        return status;
    else
        return 0;
}
int initmouse()
{
    i.x.ax=0;
    int86(0X33,&i,&o);
    return ( o.x.ax );
}
int showmouseptr()
{
    i.x.ax=1;
    int86(0X33,&i,&o);
    return 1;
}
int hidemouseptr()
{
    i.x.ax=2;
    int86(0x33,&i,&o);
    return 1;
}
int restrictmouseptr(int x1,int y1,int x2,int y2)
{
    i.x.ax=7;
    i.x.cx=x1;
    i.x.dx=x2;
    int86(0x33,&i,&o);
    i.x.ax=8;
    i.x.cx=y1;
    i.x.dx=y2;
    int86(0x33,&i,&o);
    return 1;
}
int getmousepos(int *button,int *x,int *y)
{
    i.x.ax=3;
    int86(0x33,&i,&o);
```

```
    *button=o.x.bx;
    *x=o.x.cx;
    *y=o.x.dx;
    return 1;
}
```

SCREENSHOTS



Main Menu

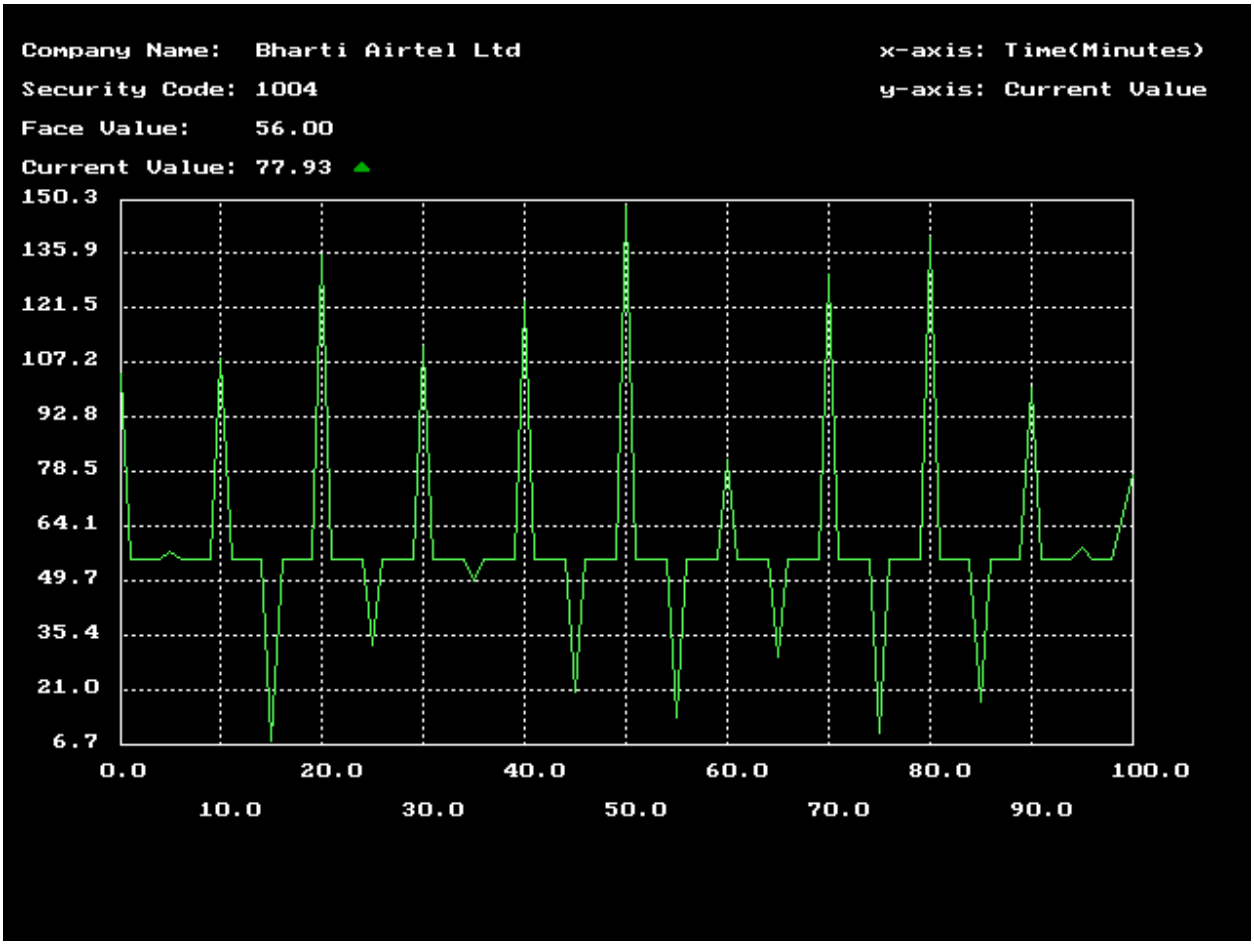

User menu
Enter 1 for displaying profile:
Enter 2 for editing your profile:
Enter 3 to manage stocks:
Enter 4 to view portfolio:
Enter 5 to view your account balance:
Enter 6 to logout:

User Menu

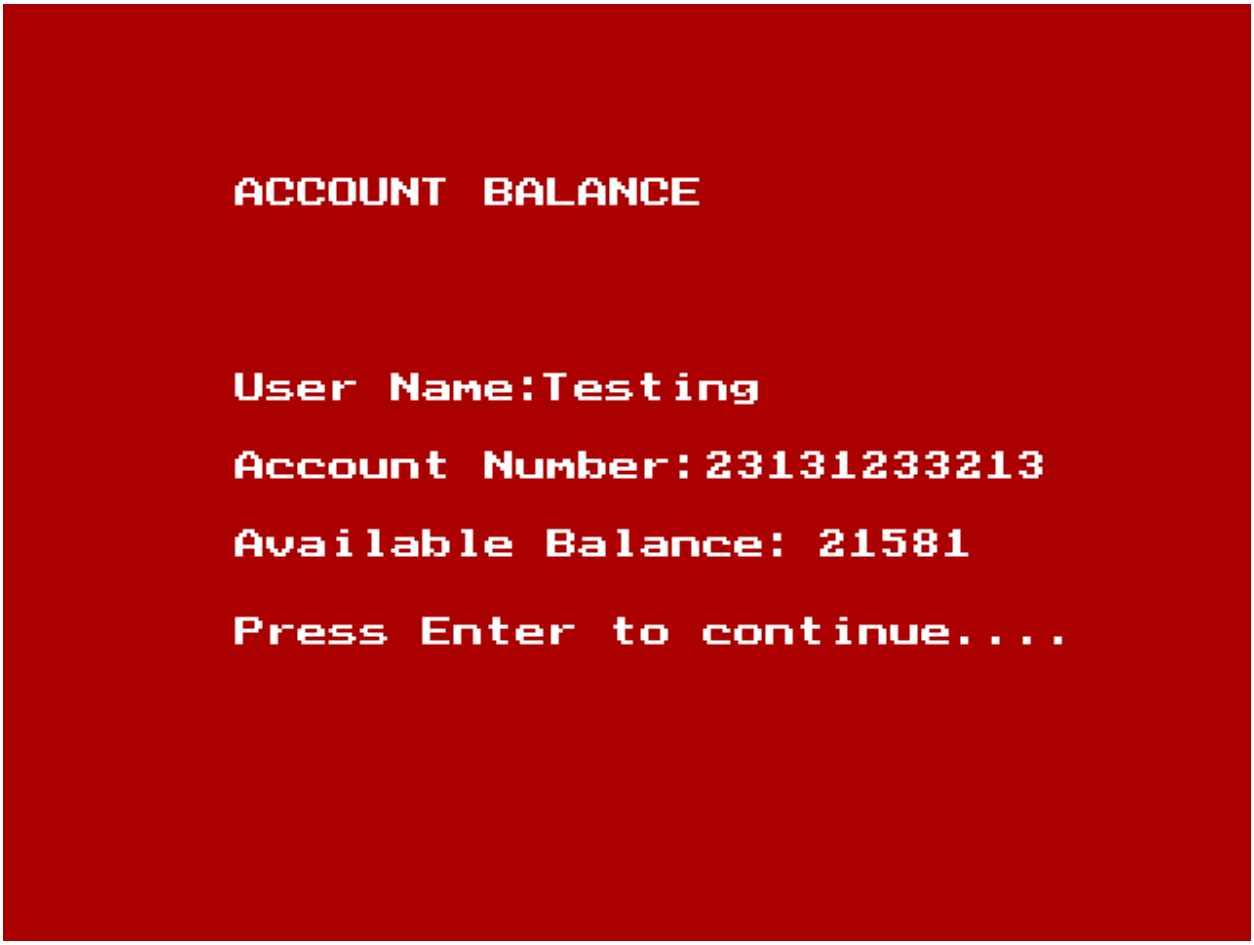
Profile Display

```
Company name: AXIS Bank
Code: 1001
Face value of each share: 34
Total shares available: 82
Current value of each share: 34.01
Increase: +0.01
Percentage increase: +0.04%
Press Enter to continue....

Company name: Bajaj Auto Ltd
Code: 1002
Face value of each share: 76
Total shares available: 96
Current value of each share: 77.08
Increase: +1.08
Percentage increase: +1.4%
Press Enter to continue....

Company name: Bharat Heavy Electricals Ltd
Code: 1003
Face value of each share: 89
Total shares available: 17
Current value of each share: 89.12
Increase: +0.12
Percentage increase: +0.13%
Press Enter to continue....
```

List of Companies



Stock Price-Time Graph of a Company



Account Balance

APPLICATION AND END USER

This program is a simplified version of an actual stock exchange. This program can be used to buy or sell stocks and make profit.

A modified version of this program can be used by stock brokers and investors who want to invest in the stock market.

BIBILOGRAPHY

- www.google.com
- Computer Science with C++ - Sumita Arora
- Graphics in C++
- www.stackoverflow.com
- www.bseindia.com