



Technical Analysis Stock Screener

Low Level Design – High Level Design

History:

Technical Analysis Stock Screener				
Date	Version	Author	Brief Description of Changes	Approver
28-09-2022	1.0	Group-3		

Index

1. Introduction	3
1.1 Intended audience	3
1.2 Project purpose	3
1.3 Key project objective	3
1.4 Project scope	4
2. Design overview	4
2.1 Design objective	5
2.2 Design alternative	5
2.3 User interface paradigms	5
2.4 Validations	5
2.5 System Requirements	5
3. Detailed System Design	6
3.1 Flowchart of main application	7
4. Functional overview	9
4.1 Header files Included	9
4.2 Functions Included	9
5. Coding	10-14
6. Tools used	15
6.1 Make file	15
6.2 Valgrind	16
6.3 Splint	17
7. Testing	18
7.1 Integration testing	18

1. Introduction

The Technical Analysis Stock Screener is a system that allows the user to enter into the system and check the available listed Stocks and their recommendations. Here the recommendations are done based on Technical Analysis. The user can see the recommendation based on those suggestions the user can do any of these like Buy, Sell and Hold the Stocks.

1.1 Intended Audience: -

The target audience set for this project can be identified as Equity Stock traders / Investors.

1.2 Project Purpose: -

The Technical Analysis Stock Screener is a project that helps us understand the basic concepts of functions like file handling, and data structure. Here this project mainly displays the list of stocks to the user and give recommendations based on the averages calculated between the latest 10 days average stock to 50 days average stock. Based on these averages It provides the recommendations like Buy, Sell and Hold of the stocks.

1.3 Key Project Objectives: -

- a. Allow the User to enter the System
- b. Can see the list of available stocks
- c. updating Functionality upto latest 20 stocks
- d. Displays all the records of The stock
- e. calculating the Average close Stocks.
- f. Suggest the recommendations like Buy, Sell or Hold.

1.4 Project scope : -

This project aims to create the development of an Technical Analysis Stock Screener ,Which takes the Stock information from various Stocks, adds it to the database and processes the Average of latest Stock records, And give recommendations based on the technical Analysis.

2. Design Overview: -

- **Technical Analysis Stock Screener Comprises of the following modules:**

Name of the Module	List of Stocks Module
Handled by	
Description	It display list of stocks available

Name of the Module	Main Menu Module
Handled by	
Description	Consist of List of choices and exit option

Name of the Module	Sub Menu Module
Handled by	
Description	It consist the information about selected choice

Name of the Module	Exit Module
Handled by	
Description	It returns to the main menu from sub menu

2.1 Design Objectives:

- Add No of stocks into the data.
- User can select which stock data want to see.
- It will Calculate averages of the stock records.
- Based on average it show recommendations to the user.
- Recommendations are buy, sell and hold.
- Then user can take decision based on technical Analysis of stocks.

2.2 Design Alternative: -

We have used a linked list structure to store data i.e.. Stock data files, to find averages and for recommendations

2.3 User Interface Paradigms: -

This project Directly perform the Technical analysis on stocks and give recommendations to the user based on the analysis.

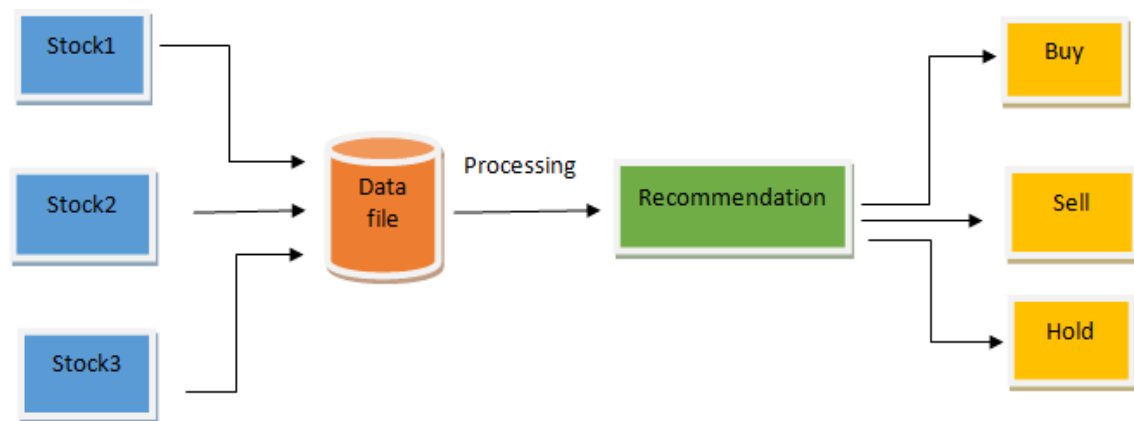
2.4 Validation: -

- User cannot enter more than one choice at a time
- User cannot perform any operations in the stocks data.

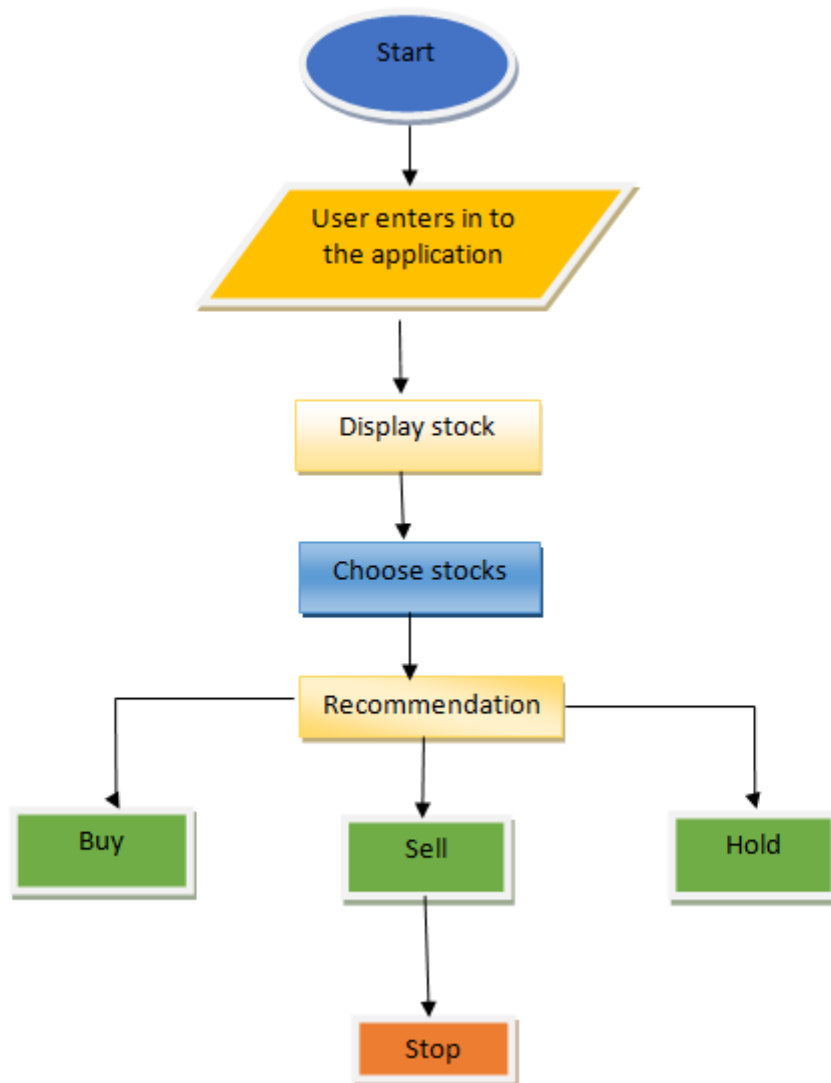
2.5 System Requirements:-

- Language Used: C language
- Tools Used : Make, Valgrind
- Compiler : Gcc
- Environment : Linux

3.DETAILED SYSTEM DESIGN:



3.1 Flow Chart of the application



4.Functional overview:

4.1 Following Header files are include:

- `#include<stdio.h>`
- `#include<stdlib.h>`
- `#include<string.h>`
- `#include<main.h>`
- `#include<struct.h>`

4.2 Following Functions are included:

A. `selectChoice()` :

This Function mainly used to give the choices to the user so that based on the list of choices the user can select from it.

B. `loadStockData()` :

This function is used to load the CSV file into the linked list. By using the file pointer we read the CSV file into the linked list.

C. `showRecommendations()` :

This Function is used to give recommendations to the stock holders based on the average calculated. It give recommendations like buy, sell or hold.

5.Coding:

Main.h:

```
sushi@sushi-HP-Notebook: ~/Desktop/project/include
```

```
#pragma once  
#include <stdlib.h>  
#include <struct.h>  
  
// Forward declarations of the functions  
  
extern void selectChoice(FILE** fptr);  
extern void loadStockData(FILE** fptr);  
extern void calculateAverage(struct NODE* head);  
  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```

1.1 All

Struct.h:

```
sushi@sushi-HP-Notebook: ~/Desktop/project/include
```

```
#pragma once
#define DATELEN 100
#define OPENLEN 100
#define HIGHLEN 100
#define LOWLEN 100
#define CLOSELEN 100
#define ADJCLOSELEN 100
#define VOLUMELEN 100
#define PARSELEN 200
#define RANGE1 (float)10
#define RANGE2 (float)50

struct NODE{

char date[DATELEN];
char open[OPENLEN];
char high[HIGHLEN];
char low[LOWLEN];
char close[CLOSELEN];
char adjClose[ADJCLOSELEN];
char volume[VOLUMELEN];
struct NODE *next;

}

1 change; before #1 20 seconds ago
```

SelectChoice ():

```
sushi@sushi-HP-Notebook: ~/Desktop/project/src
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <main.h>
#include <struct.h>

void selectChoice(FILE** fptr){
    while(1){
        system("clear");
        fflush(stdin);
        int choice;

        printf("\nList of Stocks\n\n1. AJANTA PHARMA\n2. BAJAJ ELECTRICALS\n3. CIPLA\n4. GODREJ PROP.\n5. GOODYEAR\n");
        printf("6. HAVELLES\n7. HDFC BANK\n8. MUTHOOT FIN.\n9. ONGC\n10. RAYMOND\n11. Exit\nenter your choice:");
        scanf("%d", &choice);
        switch (choice)
        {
            case 1:
                printf("\nAJANTA PHARMA\n\n");
                *fptr= fopen("../data/ajanta.csv","r");
                return;
            case 2:
                printf("\nBAJAJ ELECTRICALS\n\n");
                *fptr= fopen("../data/bajaj.csv","r");
                return;
            case 3:
                printf("\nCIPLA\n\n");
                *fptr= fopen("../data/cipla.csv","r");
                return;
            case 4:
                printf("\nGodrej Prop.\n\n");
                *fptr= fopen("../data/godrej.csv","r");
                return;
            case 5:
                printf("\nGOODYEAR\n\n");
                *fptr= fopen("../data/goodyear.csv","r");
                return;
        }
    }
}
```

1,1 Top

```
return;
case 4:
    printf("\nGodrej Prop.\n\n");
    *fptr= fopen("../data/godrej.csv","r");
    return;
case 5:
    printf("\nGOODYEAR\n\n");
    *fptr= fopen("../data/goodyear.csv","r");
    return;
case 6:
    printf("\nHAVELLES\n\n");
    *fptr= fopen("../data/havelles.csv","r");
    return;
case 7:
    printf("\nHDFC BANK\n\n");
    *fptr= fopen("../data/hdfc.csv","r");
    return;
case 8:
    printf("\nMUTHOOT FIN.\n\n");
    *fptr= fopen("../data/muthoot.csv","r");
    return;
case 9:
    printf("\nONGC\n\n");
    *fptr= fopen("../data/ongc.csv","r");
    return;
case 10:
    printf("\nRAYMOND\n\n");
    *fptr= fopen("../data/raymond.csv","r");
    return;
case 11:
    exit(0);
default:
    printf("Invalid Selection");
    break;
}
}
```

66,1 Bot

loadStockData ():

```
sushi@sushi-HP-Notebook: ~/Desktop/project/src

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <main.h>
#include <struct.h>

struct NODE *head=NULL;
struct NODE *node=NULL;

void loadStockData(FILE** fptr){

    char parsedLine[PARSEDLEN];
    int row = 0;
    while(fgets(parsedLine,PARSEDLEN,*fptr) != NULL){
        row++;
        //remove this to retain date from csv file
        if(row == 1){
            continue;
        }
        node = (struct NODE*)malloc(sizeof(struct NODE));

        char* lDate = strtok(parsedLine, ",");
        strncpy(node->date, lDate,DATELEN-1); //Datelen-1
        char *lOpen = strtok(NULL, ",");
        strncpy(node->open, lOpen,OPENLEN-1);
        char *lHigh = strtok(NULL, ",");
        strncpy(node->high, lHigh,HIGHLEN-1);
        char *lLow = strtok(NULL, ",");
        strncpy(node->low, lLow,LOWLEN-1);
        char *lClose = strtok(NULL, ",");
        strncpy(node->close, lClose,CLOSELEN-1);
        char *lAdjClose = strtok(NULL, ",");
        strncpy(node->adjClose, lAdjClose,ADJCLOSELEN-1);
        char *lVolume = strtok(NULL, ",");
        strncpy(node->volume, lVolume,VOLUMELEN-1);
        node -> next = head;
        head = node;
    }
}
```

1,1 Top

```
sushi@sushi-HP-Notebook: ~/Desktop/project/src

        char* lDate = strtok(parsedLine, ",");
        strncpy(node->date, lDate,DATELEN-1); //Datelen-1
        char *lOpen = strtok(NULL, ",");
        strncpy(node->open, lOpen,OPENLEN-1);
        char *lHigh = strtok(NULL, ",");
        strncpy(node->high, lHigh,HIGHLEN-1);
        char *lLow = strtok(NULL, ",");
        strncpy(node->low, lLow,LOWLEN-1);
        char *lClose = strtok(NULL, ",");
        strncpy(node->close, lClose,CLOSELEN-1);
        char *lAdjClose = strtok(NULL, ",");
        strncpy(node->adjClose, lAdjClose,ADJCLOSELEN-1);
        char *lVolume = strtok(NULL, ",");
        strncpy(node->volume, lVolume,VOLUMELEN-1);
        node -> next = head;
        head = node;
    }
    // fclose(*fptr);

    struct NODE* avghead= head;
    printf("Date\tOpen\tHigh\tLow\tClose\tAdj Close\tVolume\n\n");
    while(head!=NULL){
        printf("%s\t%s\t%s\t%s\t%s\t%s\t%d\n",head->date,head->open,head->high,head->low,head->close,head->adjClose,atoi(head->volume));
        head=head->next;
    }

    calculateAverage(avghead);

    struct NODE *current = NULL;
    while ((current = avghead) != NULL)
    {
        avghead = avghead->next;
        free(current);
    }
}
```

57,1 Bot

showRecommendations ():

```
sushi@sushi-HP-Notebook: ~/Desktop/project/src

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <main.h>
#include <struct.h>

void calculateAverage(struct NODE* head){
    int i;
    float sum1 = 0.0, sum2 = 0.0;
    for(i=0; i<RANGE2 && head->next != NULL; i++){
        if(i<RANGE1){
            sum1 += atof(head->adjClose);
        }
        sum2 += atof(head->adjClose);
        head = head->next;
    }
    float avg1=sum1/RANGE1;
    float avg2=sum2/RANGE2;

    if(avg1<avg2){
        printf("%f is less than %f\n", avg1, avg2);
        printf("Recommendation : BUY\n");
    }else if(avg1>avg2){
        printf("%f is greater than %f\n", avg1, avg2);
        printf("Recommendation : SELL\n");
    }else{
        printf("Recommendation : HOLD\n");
    }
}

~
~
~
~
~
~
~
~
~
~

"calculateAvg.c" [noeol] 29L, 727B 1,1 All
```


List of stocks:

```
sushi@sushi-HP-Notebook: ~/Desktop/project/bin

List of Stocks
1. AJANTA PHARMA
2. BAJAJ ELECTRICALS
3. CIPLA
4. GODREJ PROP.
5. GOODYEAR
6. HAVELLES
7. HDFC BANK
8. MUTHOOT FIN.
9. ONGC
10. RAYMOND
11. Exit
enter your choice:2

BAJAJ ELECTRICALS

Date          Open          High          Low          Close          Adj Close          Volume
2022-10-21    1168.000000    1199.500000    1151.000000    1160.800049    1160.800049    84006
2022-10-20    1158.400024    1182.849976    1153.099976    1173.400024    1173.400024    58487
2022-10-19    1154.599976    1189.949951    1152.000000    1163.400024    1163.400024    77711
2022-10-18    1146.150024    1193.099976    1140.000000    1152.750000    1152.750000    154030
2022-10-17    1133.900024    1162.599976    1133.900024    1151.099976    1151.099976    80136
2022-10-14    1162.800049    1172.000000    1138.000000    1144.300049    1144.300049    47949
2022-10-13    1155.000000    1169.949951    1141.199951    1156.699951    1156.699951    58788
2022-10-12    1189.949951    1198.550049    1156.000000    1161.400024    1161.400024    65882
2022-10-11    1180.000000    1201.949951    1166.650024    1190.400024    1190.400024    119813
2022-10-10    1192.900024    1229.000000    1160.300049    1183.900024    1183.900024    228857
2022-10-07    1188.099976    1212.000000    1177.550049    1203.849976    1203.849976    95735
2022-10-06    1188.949951    1210.000000    1177.949951    1188.300049    1188.300049    119685
2022-10-04    1204.699951    1213.250000    1160.099976    1189.050049    1189.050049    141657
2022-10-03    1198.400024    1229.900024    1177.449951    1191.000000    1191.000000    204811
2022-09-30    1153.849976    1216.900024    1124.550049    1202.199951    1202.199951    365305
2022-09-29    1096.800049    1150.000000    1095.900024    1136.349976    1136.349976    128070
2022-09-28    1073.900024    1108.000000    1060.000000    1091.099976    1091.099976    59126
2022-09-27    1079.949951    1088.000000    1035.550049    1074.500000    1074.500000    78043
```

Output:

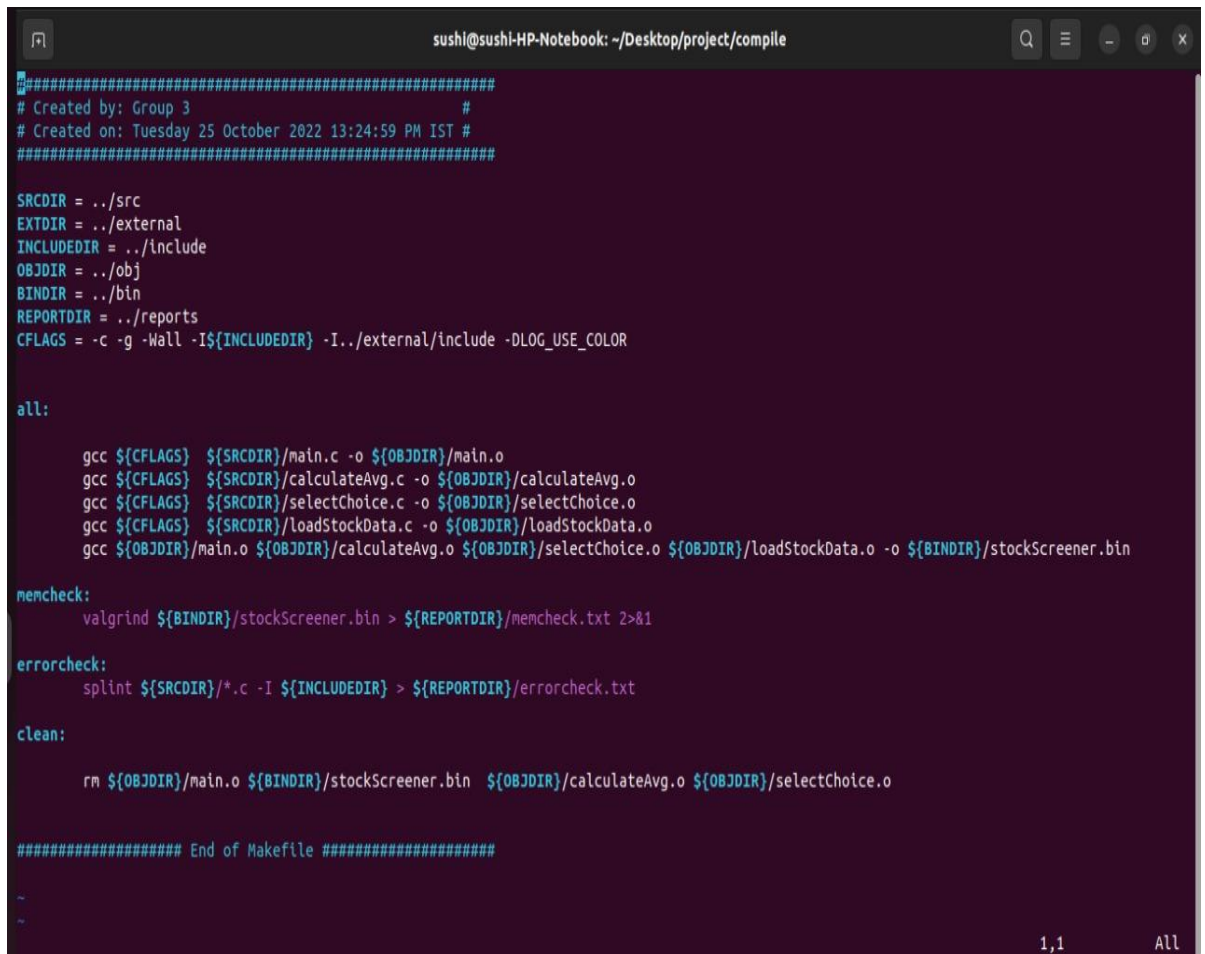
```
sushi@sushi-HP-Notebook: ~/Desktop/project/bin

2022-09-09    1214.500000    1217.500000    1199.150024    1204.449951    1204.449951    51228
2022-09-08    1236.000000    1238.199951    1200.199951    1208.300049    1208.300049    103273
2022-09-07    1229.650024    1255.000000    1213.949951    1226.900024    1226.900024    173494
2022-09-06    1234.150024    1248.400024    1225.000000    1229.650024    1229.650024    80109
2022-09-05    1223.900024    1242.699951    1198.000000    1224.250000    1224.250000    59201
2022-09-02    1217.650024    1230.000000    1200.150024    1222.250000    1222.250000    108188
2022-09-01    1219.550049    1243.800049    1195.000000    1209.949951    1209.949951    86181
2022-08-30    1216.949951    1237.800049    1211.900024    1232.900024    1232.900024    124495
2022-08-29    1214.900024    1230.000000    1194.000000    1212.000000    1212.000000    50501
2022-08-26    1253.000000    1264.599976    1210.050049    1224.000000    1224.000000    176242
2022-08-25    1231.000000    1260.000000    1231.000000    1246.449951    1246.449951    138163
2022-08-24    1250.000000    1259.050049    1216.650024    1230.099976    1230.099976    127820
2022-08-23    1245.000000    1265.000000    1233.000000    1247.900024    1247.900024    174916
2022-08-22    1228.349976    1262.000000    1203.150024    1245.800049    1245.800049    353584
2022-08-19    1240.000000    1250.000000    1203.250000    1228.349976    1228.349976    270621
2022-08-18    1165.000000    1240.000000    1155.800049    1226.150024    1226.150024    1099035
2022-08-17    1160.000000    1183.599976    1145.300049    1155.750000    1155.750000    177918
2022-08-16    1145.300049    1174.000000    1130.949951    1142.599976    1142.599976    69865
2022-08-12    1157.650024    1186.000000    1134.000000    1143.550049    1143.550049    349096
2022-08-11    1169.900024    1169.900024    1145.300049    1156.400024    1156.400024    35965
2022-08-10    1140.000000    1173.699951    1119.250000    1161.400024    1161.400024    147605
2022-08-08    1135.650024    1185.000000    1125.099976    1150.550049    1150.550049    137005
2022-08-05    1146.050049    1148.000000    1125.000000    1131.050049    1131.050049    21972
2022-08-04    1129.000000    1155.000000    1113.099976    1141.949951    1141.949951    99562
2022-08-03    1140.000000    1140.000000    1120.199951    1127.400024    1127.400024    21817
2022-08-02    1137.000000    1140.250000    1122.349976    1132.000000    1132.000000    26384
2022-08-01    1145.199951    1160.000000    1133.900024    1139.650024    1139.650024    23234
2022-07-29    1160.000000    1160.000000    1126.050049    1133.150024    1133.150024    58370
2022-07-28    1130.000000    1149.400024    1105.650024    1113.800049    1113.800049    39529
2022-07-27    1133.349976    1144.650024    1126.199951    1129.349976    1126.349976    24933
2022-07-26    1181.000000    1183.849976    1125.800049    1133.349976    1130.339355    51901
2022-07-25    1165.000000    1179.000000    1155.349976    1169.000000    1165.894653    67424
2022-07-22    1145.000000    1156.599976    1125.199951    1153.599976    1150.535522    81910
1163.815063 is less than 1176.741821
Recommendation : BUY
Press 1:For Main Menu
Press 0:To Exit Program
```

6.Tools Used:

1.Make file:

Make file is a set of commands with variable names and targets to create object files and to remove them. In a single makefile we can create multiple targets to compile and to remove object, binary files. We can compile our project any number of times using make file.



```
#####
# Created by: Group 3                                     #
# Created on: Tuesday 25 October 2022 13:24:59 PM IST #
#####

SRCDIR = ../src
EXTDIR = ../external
INCLUDEDIR = ../include
OBJDIR = ../obj
BINDIR = ../bin
REPORTDIR = ../reports
CFLAGS = -c -g -Wall -I${INCLUDEDIR} -I../external/include -DLOG_USE_COLOR

all:

gcc ${CFLAGS} ${SRCDIR}/main.c -o ${OBJDIR}/main.o
gcc ${CFLAGS} ${SRCDIR}/calculateAvg.c -o ${OBJDIR}/calculateAvg.o
gcc ${CFLAGS} ${SRCDIR}/selectChoice.c -o ${OBJDIR}/selectChoice.o
gcc ${CFLAGS} ${SRCDIR}/loadStockData.c -o ${OBJDIR}/loadStockData.o
gcc ${OBJDIR}/main.o ${OBJDIR}/calculateAvg.o ${OBJDIR}/selectChoice.o ${OBJDIR}/loadStockData.o -o ${BINDIR}/stockScreener.bin

memcheck:
    valgrind ${BINDIR}/stockScreener.bin > ${REPORTDIR}/memcheck.txt 2>&1

errorcheck:
    splint ${SRCDIR}/*.c -I ${INCLUDEDIR} > ${REPORTDIR}/errorcheck.txt

clean:

rm ${OBJDIR}/main.o ${BINDIR}/stockScreener.bin ${OBJDIR}/calculateAvg.o ${OBJDIR}/selectChoice.o

##### End of Makefile #####

~
~
1,1 All
```

2.Valgrind:

```
sushi@sushi-HP-Notebook: ~/Desktop/project/reports
==36660== Memcheck, a memory error detector
==36660== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==36660== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==36660== Command: ../bin/stockScreener.bin
==36660==
^[[H^[[2J^[[3J
List of Stocks

1.AJANTA PHARMA
2.BAJAJ ELECTRICALS
3.CIPLA
4.GODREJ PROP.
5.GOODYEAR
6.HAVELLES
7.HDFC BANK
8.MUTHOOT FIN.
9.ONGC
10.RAYMOND
11.Exit
enter your choice:
BAJAJ ELECTRICALS

Date          Open          High          Low          Close          Adj Close          Volume
2022-10-21    1168.000000    1199.500000    1151.000000    1160.800049    1160.800049    84006
2022-10-20    1158.400024    1182.849976    1153.099976    1173.400024    1173.400024    58487
2022-10-19    1154.599976    1189.949951    1152.000000    1163.400024    1163.400024    77711
2022-10-18    1146.150024    1193.099976    1140.000000    1152.750000    1152.750000    154030
2022-10-17    1133.900024    1162.599976    1133.900024    1151.099976    1151.099976    80136
2022-10-14    1162.800049    1172.000000    1138.000000    1144.300049    1144.300049    47949
2022-10-13    1155.000000    1169.949951    1141.199951    1156.699951    1156.699951    58788
2022-10-12    1189.949951    1198.550049    1156.000000    1161.400024    1161.400024    65882
2022-10-11    1180.000000    1201.949951    1166.650024    1190.400024    1190.400024    119813
2022-10-10    1192.900024    1229.000000    1160.300049    1183.900024    1183.900024    228857
2022-10-07    1188.099976    1212.000000    1177.550049    1203.849976    1203.849976    95735
2022-10-06    1188.949951    1210.000000    1177.949951    1188.300049    1188.300049    119685
2022-10-04    1204.699951    1213.250000    1160.099976    1189.050049    1189.050049    141657

1,1 Top
```

```
sushi@sushi-HP-Notebook: ~/Desktop/project/reports
2022-08-30    1216.949951    1237.800049    1211.900024    1232.900024    1232.900024    124495
2022-08-29    1214.900024    1230.000000    1194.000000    1212.000000    1212.000000    50501
2022-08-26    1253.000000    1264.599976    1210.050049    1224.000000    1224.000000    176242
2022-08-25    1231.000000    1260.000000    1231.000000    1246.449951    1246.449951    138163
2022-08-24    1250.000000    1259.050049    1216.650024    1230.099976    1230.099976    127820
2022-08-23    1245.000000    1265.000000    1233.000000    1247.900024    1247.900024    174916
2022-08-22    1228.349976    1262.000000    1203.150024    1245.800049    1245.800049    353584
2022-08-19    1240.000000    1250.000000    1203.250000    1228.349976    1228.349976    270621
2022-08-18    1165.000000    1240.000000    1155.800049    1226.150024    1226.150024    1099035
2022-08-17    1160.000000    1183.599976    1145.300049    1155.750000    1155.750000    177918
2022-08-16    1145.300049    1174.000000    1130.949951    1142.599976    1142.599976    69865
2022-08-12    1157.650024    1186.000000    1134.000000    1143.550049    1143.550049    349096
2022-08-11    1169.900024    1169.900024    1145.300049    1156.400024    1156.400024    35965
2022-08-10    1140.000000    1173.699951    1119.250000    1161.400024    1161.400024    147605
2022-08-08    1135.650024    1185.000000    1125.099976    1150.550049    1150.550049    137005
2022-08-05    1146.050049    1148.000000    1125.000000    1131.050049    1131.050049    21972
2022-08-04    1129.000000    1155.000000    1113.099976    1141.949951    1141.949951    99562
2022-08-03    1140.000000    1140.000000    1120.199951    1127.400024    1127.400024    21817
2022-08-02    1137.000000    1140.250000    1122.349976    1132.000000    1132.000000    26384
2022-08-01    1145.199951    1160.000000    1133.900024    1139.650024    1139.650024    23234
2022-07-29    1160.000000    1160.000000    1126.050049    1133.150024    1133.150024    58370
2022-07-28    1130.000000    1149.400024    1105.650024    1113.800049    1113.800049    39529
2022-07-27    1133.349976    1144.650024    1126.199951    1129.349976    1126.349976    24933
2022-07-26    1181.000000    1183.849976    1125.800049    1133.349976    1130.339355    51901
2022-07-25    1165.000000    1179.000000    1155.349976    1169.000000    1165.894653    67424
2022-07-22    1145.000000    1156.599976    1125.199951    1153.599976    1150.535522    81910

Recommendation : BUY
Press 1:For Main Menu
Press 0:To Exit Program==36660==
==36660== HEAP SUMMARY:
==36660==    in use at exit: 0 bytes in 0 blocks
==36660==    total heap usage: 65 allocs, 65 frees, 52,808 bytes allocated
==36660== All heap blocks were freed -- no leaks are possible
==36660==
==36660== For lists of detected and suppressed errors, rerun with: -s
==36660== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)

97,1 Bot
```


3.Splint:

```
sushi@sushi-HP-Notebook: ~/Desktop/project/reports

../src/loadStockData.c:7:19: Global head initialized to null value: head = NULL
A reference with no null annotation is assigned or initialized to NULL. Use
/*@null@*/ to declare the reference as a possibly null pointer. (Use
-nullassign to inhibit warning)
../src/loadStockData.c:7:19: Global head initialized to null value:
    struct NODE * head = NULL = NULL
../src/loadStockData.c:8:19: Global node initialized to null value: node = NULL
../src/loadStockData.c:8:19: Global node initialized to null value:
    struct NODE * node = NULL = NULL
../src/loadStockData.c: (in function loadStockData)
../src/loadStockData.c:22:20: Arrow access from possibly null pointer node:
    node->date
A possibly null pointer is dereferenced. Value is either the result of a
function which may return null (in which case, code should check it is not
null), or a global, parameter or structure field declared with the null
qualifier. (Use -nullderefer to inhibit warning)
../src/loadStockData.c:20:16: Storage node may become null
../src/loadStockData.c:24:28: Possibly null storage getOpen passed as non-null
    param: strcpy (... , getOpen)
A possibly null pointer is passed as a parameter corresponding to a formal
parameter with no /*@null@*/ annotation. If NULL may be used for this
parameter, add a /*@null@*/ annotation to the function parameter declaration.
(Use -nullpass to inhibit warning)
../src/loadStockData.c:23:25: Storage getOpen may become null
../src/loadStockData.c:26:28: Possibly null storage getHigh passed as non-null
    param: strcpy (... , getHigh)
../src/loadStockData.c:25:25: Storage getHigh may become null
../src/loadStockData.c:28:27: Possibly null storage getLow passed as non-null
    param: strcpy (... , getLow)
../src/loadStockData.c:27:24: Storage getLow may become null
../src/loadStockData.c:30:29: Possibly null storage getClose passed as non-null
    param: strcpy (... , getClose)
../src/loadStockData.c:29:26: Storage getClose may become null
../src/loadStockData.c:32:32: Possibly null storage getAdjClose passed as
non-null param: strcpy (... , getAdjClose)
../src/loadStockData.c:31:29: Storage getAdjClose may become null
../src/loadStockData.c:34:30: Possibly null storage getVolume passed as
```

1,1 Top

7. Testing:

7.1 Integration testing:

Test_case 1 (Recommendation Buy):

```
sushi@sushi-HP-Notebook: ~/Desktop/project/bin

2022-09-09 1214.500000 1217.500000 1199.150024 1204.449951 1204.449951 51228
2022-09-08 1236.000000 1238.199951 1200.199951 1208.300049 1208.300049 103273
2022-09-07 1229.650024 1255.000000 1213.949951 1226.900024 1226.900024 173494
2022-09-06 1234.150024 1248.400024 1225.000000 1229.650024 1229.650024 80109
2022-09-05 1223.900024 1242.699951 1198.000000 1224.250000 1224.250000 59201
2022-09-02 1217.650024 1230.000000 1200.150024 1222.250000 1222.250000 100188
2022-09-01 1219.550049 1243.800049 1195.000000 1209.949951 1209.949951 86181
2022-08-30 1216.949951 1237.800049 1211.900024 1232.900024 1232.900024 124495
2022-08-29 1214.900024 1230.000000 1194.000000 1212.000000 1212.000000 50501
2022-08-26 1253.000000 1264.599976 1210.050049 1224.000000 1224.000000 176242
2022-08-25 1231.000000 1260.000000 1231.000000 1246.449951 1246.449951 138163
2022-08-24 1250.000000 1259.050049 1216.650024 1230.099976 1230.099976 127820
2022-08-23 1245.000000 1265.000000 1233.000000 1247.900024 1247.900024 174916
2022-08-22 1228.349976 1262.000000 1203.150024 1245.800049 1245.800049 353584
2022-08-19 1240.000000 1250.000000 1203.250000 1228.349976 1228.349976 270621
2022-08-18 1165.000000 1240.000000 1155.800049 1226.150024 1226.150024 1099035
2022-08-17 1160.000000 1183.599976 1145.300049 1155.750000 1155.750000 177918
2022-08-16 1145.300049 1174.000000 1130.949951 1142.599976 1142.599976 69865
2022-08-12 1157.650024 1186.000000 1134.000000 1143.550049 1143.550049 349096
2022-08-11 1169.900024 1169.900024 1145.300049 1156.400024 1156.400024 35965
2022-08-10 1140.000000 1173.699951 1119.250000 1161.400024 1161.400024 147605
2022-08-08 1135.650024 1185.000000 1125.099976 1150.550049 1150.550049 137005
2022-08-05 1146.050049 1148.000000 1125.000000 1131.050049 1131.050049 21972
2022-08-04 1129.000000 1155.000000 1113.099976 1141.949951 1141.949951 99562
2022-08-03 1140.000000 1140.000000 1120.199951 1127.400024 1127.400024 21817
2022-08-02 1137.000000 1140.250000 1122.349976 1132.000000 1132.000000 26384
2022-08-01 1145.199951 1160.000000 1133.900024 1139.650024 1139.650024 23234
2022-07-29 1160.000000 1160.000000 1126.050049 1133.150024 1133.150024 58370
2022-07-28 1130.000000 1149.400024 1105.650024 1113.800049 1113.800049 39529
2022-07-27 1133.349976 1144.650024 1126.199951 1129.349976 1126.349976 24933
2022-07-26 1181.000000 1183.849976 1125.800049 1133.349976 1130.339355 51901
2022-07-25 1165.000000 1179.000000 1155.349976 1169.000000 1165.894653 67424
2022-07-22 1145.000000 1156.599976 1125.199951 1153.599976 1150.535522 81910
1163.815063 is less than 1176.741821
Recommendation : BUY
Press 1:For Main Menu
Press 0:To Exit Program
```

Test case-2(Recommendation Sell):

```
sushi@sushi-HP-Notebook: ~/Desktop/project/bin

2022-09-09 1060.000000 1065.949951 1055.000000 1058.300049 1058.300049 1092235
2022-09-08 1058.500000 1059.750000 1044.000000 1056.050049 1056.050049 1755391
2022-09-07 1033.849976 1061.849976 1033.000000 1051.500000 1051.500000 2471263
2022-09-06 1025.000000 1042.000000 1024.900024 1040.849976 1040.849976 1360910
2022-09-05 1012.500000 1030.449951 1012.000000 1025.650024 1025.650024 1275368
2022-09-02 1024.000000 1031.199951 1012.049988 1015.599976 1015.599976 1655229
2022-09-01 1026.000000 1044.250000 1019.400024 1023.000000 1023.000000 1528731
2022-08-30 1018.500000 1042.199951 1016.099976 1038.449951 1038.449951 1571042
2022-08-29 1000.000000 1024.000000 991.299988 1017.299988 1017.299988 1305557
2022-08-26 1030.900024 1035.699951 1018.500000 1020.799988 1020.799988 1264070
2022-08-25 1043.949951 1043.949951 1020.549988 1025.750000 1025.750000 1112617
2022-08-24 1041.000000 1053.849976 1026.050049 1037.349976 1037.349976 1562062
2022-08-23 1010.200012 1046.800049 1010.200012 1040.650024 1040.650024 1868332
2022-08-22 1016.000000 1029.849976 1015.500000 1022.549988 1022.549988 993053
2022-08-19 1029.050049 1036.599976 1020.099976 1030.099976 1030.099976 654004
2022-08-18 1028.500000 1034.050049 1007.349976 1032.199951 1032.199951 1201720
2022-08-17 1039.750000 1042.949951 1024.199951 1025.750000 1025.750000 1102323
2022-08-16 1027.150024 1036.000000 1025.250000 1034.550049 1034.550049 776409
2022-08-12 1043.599976 1043.599976 1022.349976 1027.150024 1027.150024 1140717
2022-08-11 1040.500000 1049.000000 1035.000000 1038.400024 1038.400024 838963
2022-08-10 1035.550049 1046.300049 1029.050049 1038.349976 1038.349976 1293683
2022-08-08 1036.199951 1039.500000 1024.099976 1029.550049 1029.550049 767289
2022-08-05 1040.000000 1045.949951 1032.000000 1034.199951 1029.199951 1561405
2022-08-04 1017.799988 1046.900024 1015.000000 1044.500000 1039.450195 2865077
2022-08-03 1010.000000 1028.750000 1004.500000 1011.849976 1006.958008 4052763
2022-08-02 1003.650024 1006.099976 992.000000 1004.200012 999.345032 1184696
2022-08-01 990.000000 1031.000000 988.299988 1004.549988 999.693359 4647912
2022-07-29 973.700012 983.950012 953.000000 977.400024 972.674622 2941566
2022-07-28 977.000000 978.950012 964.200012 967.450012 962.772705 1043837
2022-07-27 956.900024 976.250000 951.799988 974.049988 969.340820 904733
2022-07-26 959.950012 962.000000 944.500000 955.700012 951.079529 988405
2022-07-25 976.000000 979.799988 955.049988 959.799988 955.159668 1024751
2022-07-22 973.500000 976.500000 967.500000 972.099976 967.400208 977238
1117.749878 is greater than 1069.997070
Recommendation : SELL
Press 1:For Main Menu
Press 0:To Exit Program
```

