Koneru Lakshmaiah Education Foundation

(Deemed to be University)

FRESHMAN ENGINEERING DEPARTMENT

A Project Based Lab Report

On

STOCK TABLE

SUBMITTED BY:

I.D NUMBER	NAME
2200031513	P.Nitin Sai
2200031514	Rohit Kumar Singh
2200031516	E.Ashok Kumar Reddy
2200031517	U.Chetan Sai

UNDER THE GUIDANCE OF

Dr K. Arun Bhaskar

Associate Professor,

Dept of Basic Engineering & Sciences.

KL UNIVERSITY

Green fields, Vaddeswaram – 522 302

Guntur Dt., AP, India.

DEPARTMENT OF BASIC ENGINEERING SCIENCES-1

CERTIFICATE

This is to certify that the project based laboratory report entitled "Stock Table" submitted by Mr./Ms**. P.Nitin Sai, Rohit Kumar Singh, E.Ashok Kumar Reddy, U.Chetan Sai** bearing Regd. No. 2200031513, 2200031514, 2200031516, 2200031517 to the **Department of Basic Engineering Sciences-1, KL University** in partial fulfillment of the requirements for the completion of a project based Laboratory in "COMPUTATIONAL THINKING FOR STRUCTURED DESIGN"course in I B Tech I Semester, is a bonafide record of the work carried out by him/her under my supervision during the academic year 2022 – 2023.

PROJECT SUPERVISOR HEAD OF THE DEPARTMENT

Dr K. Arun Bhaskar **Dr.D.Haritha**

ACKNOWLEDGEMENTS

It is great pleasure for me to express my gratitude to our honorable President **Sri. Koneru Satyanarayana**, for giving the opportunity and platform with facilities in accomplishing the project based laboratory report.

I express the sincere gratitude to our principal **Dr. A. Jagadeesh** for his administration towards our academic growth.

I express sincere gratitude to HOD-BES-1 **Dr. D. Haritha** for her leadership and constant motivation provided in successful completion of our academic semester. I record it as my privilege to deeply thank for providing us the efficient faculty and facilities to make our ideas into reality.

I express my sincere thanks to our project supervisor **Dr K. Arun Bhaskar** for his/her novel association of ideas, encouragement, appreciation and intellectual zeal which motivated us to venture this project successfully.

Finally, it is pleased to acknowledge the indebtedness to all those who devoted themselves directly or indirectly to make this project report success.

Name: P.Nitin Sai Regd . No: 2200031513

Rohit Kumar Singh 2200031514

E.Ashok Kumar Reddy 2200031516

U.Chetan Sai 2200031517

ABSTRACT

In the era of big data, deep learning for predicting stock market prices and trends has become even more popular than before. We collected 2 years of data from Chinese stock market and proposed a comprehensive customization of feature engineering and deep learning-based model for predicting price trend of stock markets. The proposed solution is comprehensive as it includes pre-processing of the stock market dataset, utilization of multiple feature engineering techniques, combined with a customized deep learning-based system for stock market price trend prediction. We conducted comprehensive evaluations on frequently used machine learning models and conclude that our proposed solution outperforms due to the comprehensive feature engineering that we built. The system achieves overall high accuracy for stock market trend prediction. With the detailed design and evaluation of prediction term lengths, feature engineering, and data preprocessing methods, this work contributes to the stock analysis research community both in the financial and technical domains.

INDEX

INTRODUCTION

• Data Types in C:- A data-type in C programming is a set of values and is determined to act on those values. C provides various types of data-types, allowing the programmer to select the appropriate type for the variable to set its value.

The data-type in a programming language is the collection of data with values having fixed meanings and characteristics. Some of them are an integer, floating point, character, etc. Usually, programming languages specify the range values for a given datatype.

C Data Types are used to:-

- Identify the type of a variable when it is declared.
- Identify the type of return value of a function.
- Identify the type of parameter expected by a function.
- Strings:-

  Strings are actually one-dimensional array of characters terminated by a **null** character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a **null**.

  The following declaration and initialization create a string consisting of the word "Hello". To hold the null character at the end of the array, the size of the character array containing the string is one more than the number of characters in the word "Hello."

  char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};

  If you follow the rule of array initialization then you can write the above statement as follows

  char greeting[] = "Hello";

  • String Functions:-

  The following are the string functions in C:

  Function Description

  strlen():- It returns the string's length.

  strnlen():- It returns the specified value if the value specified is less than the stringlength, otherwise the string length.

  strcmp():- It compares two strings and returns 0 if the strings are the same.

  strncmp():- It compares two strings only to n characters.

  strcat():- It concatenates two strings and returns the concatenated string.

  strncat():- It concatenates n characters of one string to another string.

  strcpy():- It copies one string into another.

  strncpy():- It copies the first n characters of one string into another.

strchr() :-It finds out the first occurrence of a given character in a string. Function Description

strrchr():- It finds out the last occurrence of a given character in a string.

strstr() :-It finds out the first occurrence of a given string in a string. It finds out the first occurrence of a given string in a string where

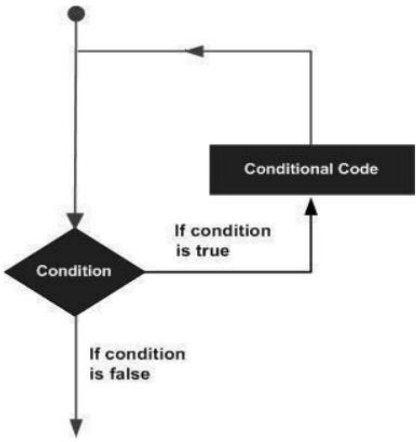
strnstr():- the searchis limited to n characters.

strcasecmp():- It compares two strings without sensitivity to the case. It compares n characters of one string to another without

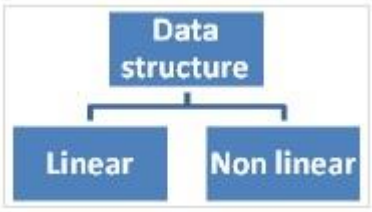
strncasecmp():- sensitivityto the case.

Loops:-

Programming languages provide various control structures that allow for more complicated execution paths. A loop statement allows us to execute a statement or group of statements multiple times. Given below is the general form of a loop statement in most of the programming languages



Data structure is a collection of data organised in a structured way. It is classified into two types which are linear data structure and non-linear data structure.



Linear data structure − Here, data is organised in a linear fashion.

For example − Arrays, structures, stacks, queues, linked lists.

Non-linear data structure − Here, Data is organised in a hierarchical way.

For example − Trees, graphs, sets, tables.

Stack in C language

It is a linear data structure, where data is inserted and removed only at one end.

Operations

- Push – Inserting an element into a stack.
- Pop – Deleting an element from a stack.

Conditions

- Stack Overflow: It is a condition that happens when we try to push more elements into the already saturated stack (Trying to insert an element into a full stack.)
- Stack Underflow: It is a condition that happens when we try to pop an element from an empty stack(Try to delete an element from a stack which is empty.)

Push ( )

- Check for stack overflow.

```
if (top = = n-1)
printf("stack over flow”);
```

- Otherwise, insert an element into the stack.

```
top ++
a[top] = item
```

Pop ( )

- Check for stack underflow.

```
if ( top = = -1)
printf( "stack under flow”);
```

- Otherwise, delete the element from the stack.

```
item = a[top]
top --
```

Display ( )

- Check for stack flow.

```
if (top == -1)
printf ("stack is empty”);
```

- Otherwise, follow the below mentioned algorithm –

```
for (i=0; i<top; i++)
printf ("%d”, a[i]);
```

Stock Table:- Reading a stock table is a way to **gain insight into the performance of a particular stock's performance**. Viewing a stock table can help you determine if you want to invest in that stock or not. However, if you are unable to properly read a stock table your decision may not be a good one.

AIM

CREATE A PROGRAM THAT SHOULD HAVE THE USER ABLE TO ENTER INFORMATION ABOUT VARIOUS STOCKS, THE AMOUNT OF SHARES, AND THE PRICE. THE USER CAN THEN ENTER A QUERY ABOUT A CERTAIN STOCK AND THE COST ACCORDING TO THE LIFO ACCOUNTING METHODS FOR A CERTAIN NUMBER OF SHARES.

- The name of the stock (a string or int)
- The number of shares of a stock (an int)
- The purchase price (can be a decimal)

The following could be your menu:

- PRESS 1 TO ENTER A NEW STOCK.
- PRESS 2 TO FIND THE LIFO PRICE FOR A STOCK.
- IF 1 IS PRESSED, THE USER NEEDS TO ENTER THE STOCK SYMBOL, AND THE NUMBER OF SHARES, AND THE PRICE.
- IF 2 IS PRESSED, THE USER NEEDS TO ENTER THE STOCK SYMBOL BEING QUERIED AND THE NUMBER OF SHARES IN QUESTION

Advantages:-

You can view details of the available stock at different locations and make a purchase or sales decision accordingly. A clear view of the purchase and sales order outstanding helps you further decide on the next steps to improve the cash flow in your business.

- It aggregates the data from different times
- It shows visual display of the results
- It helps in understanding the performance of the company
- Comparison can be made between different companies

Disadvantages:-

You can view details of the available stock at different locations and make a purchase or sales decision accordingly. A clear view of the purchase and sales order outstanding helps you further decide on the next steps to improve the cash flow in your business.

- Some time data can be inaccurate
- Because of dynamic memory allocation if we not use all memory space then there will be wastage of memory space
- Creating too many objects on the stack can increase the risk of stack over flow.
- Random access is not possible.

Future enhancements:-

In this assignment, we implemented stock table operations with below mentioned options using stack by following LIFO accounting.1. Insert a stock details2. Find price for a stock. Where each stock holds name, no. Of shares and purchase price of stock. For example: Companies and people often buy and sells stocks. Often, they buy the same stock for different prices at different times. Say a person owns 1000 shares a certain stock (such as Checkpoint), she may have bought the stock in amounts of 100 shares over 10 different times with 10different prices

SYSTEM REQUIREMENTS

SOFTWARE REQUIREMENTS:

The major software requirements of the project are as follows:

- Language : Turbo-C
- Operating system**:**Windows Xp or later.

HARDWARE REQUIREMENTS:

The hardware requirements that map towards the software are as follows:

- RAM : 1GB or 1GB plus
- Processor :Intel® Pentium® or compatible, 1.6 GHz minimum (2GHz+recommended)

ALGORITHM

STEP 1:- create two string called stock name and stock symbol , long long int for entering stock price.

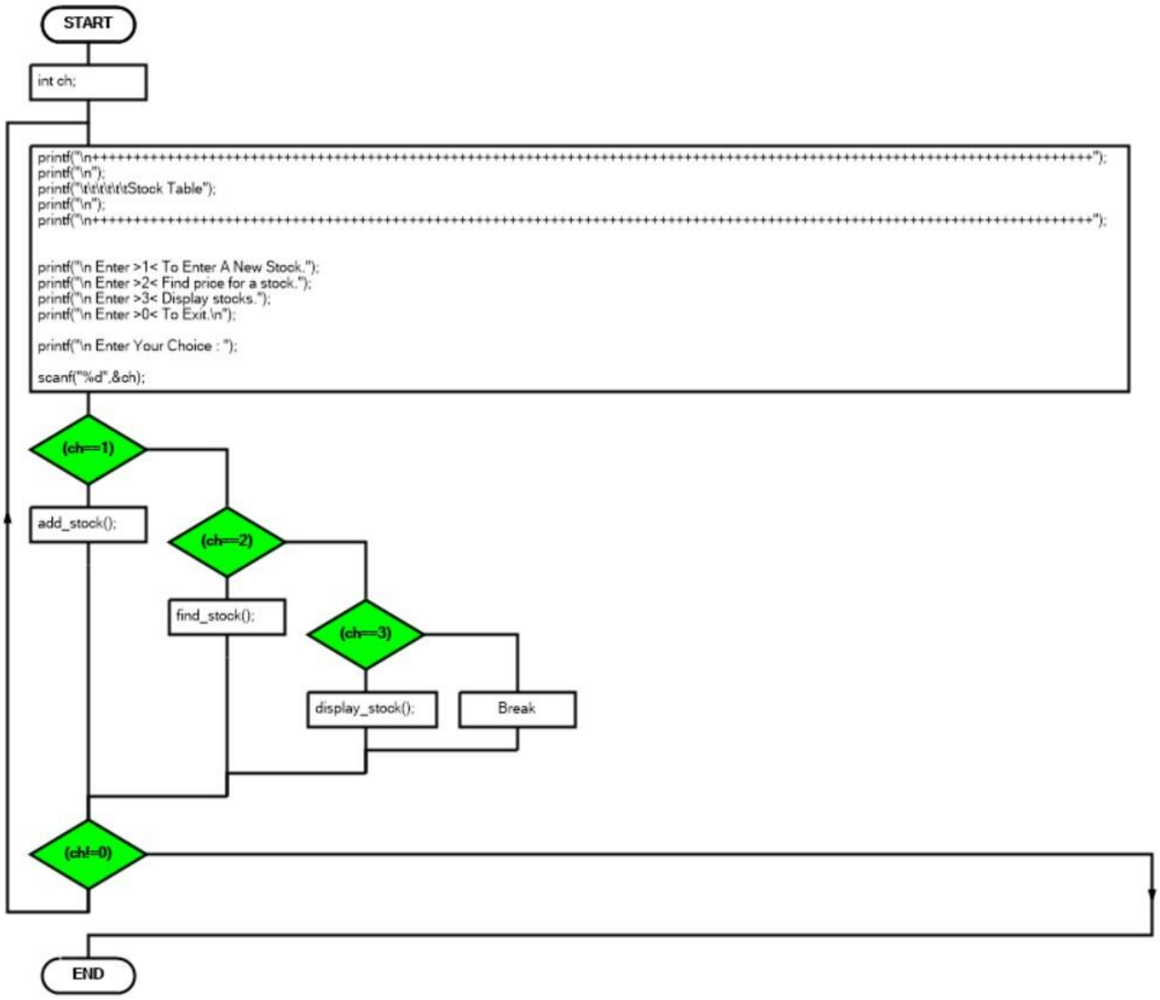
STEP 2:- using switch case if 1 is entered add_stock() function will takes place,if 2 is entered find_stock() function will takes place , if 3 is entered display() will takes place ,if 0 is entered program will be terminated.

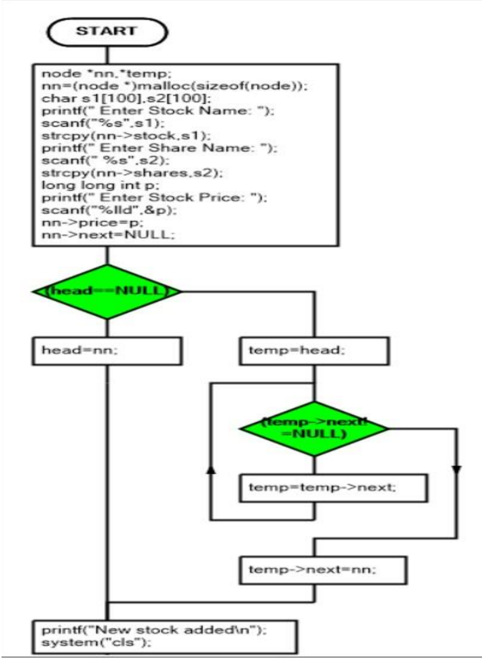
STEP 3:-in add_stock() enter the stock name and symbol and price of stock

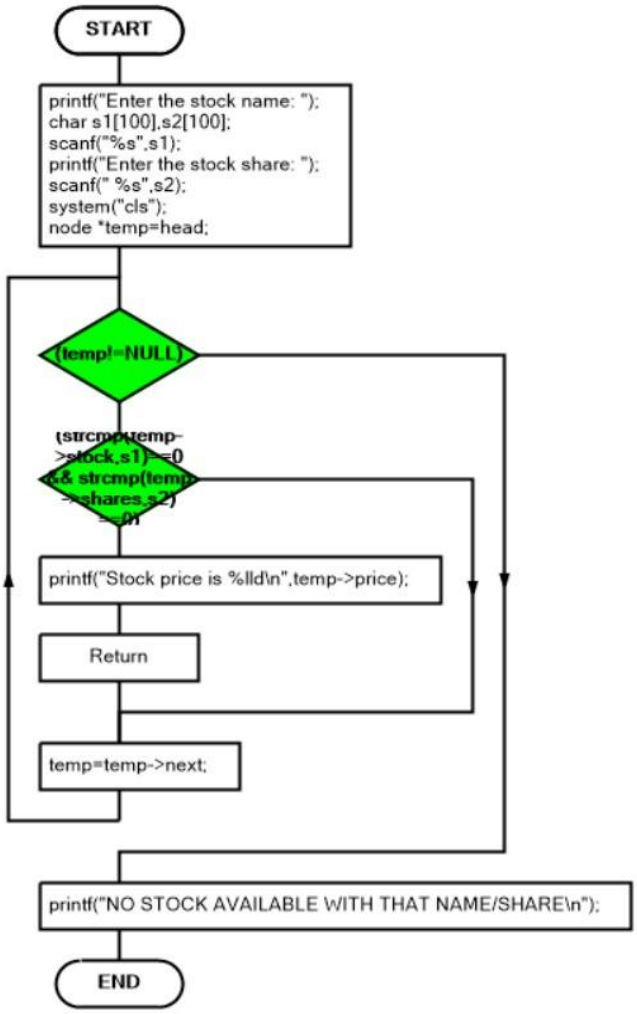
STEP 4:-in find_stock() enter the stock name and symbol

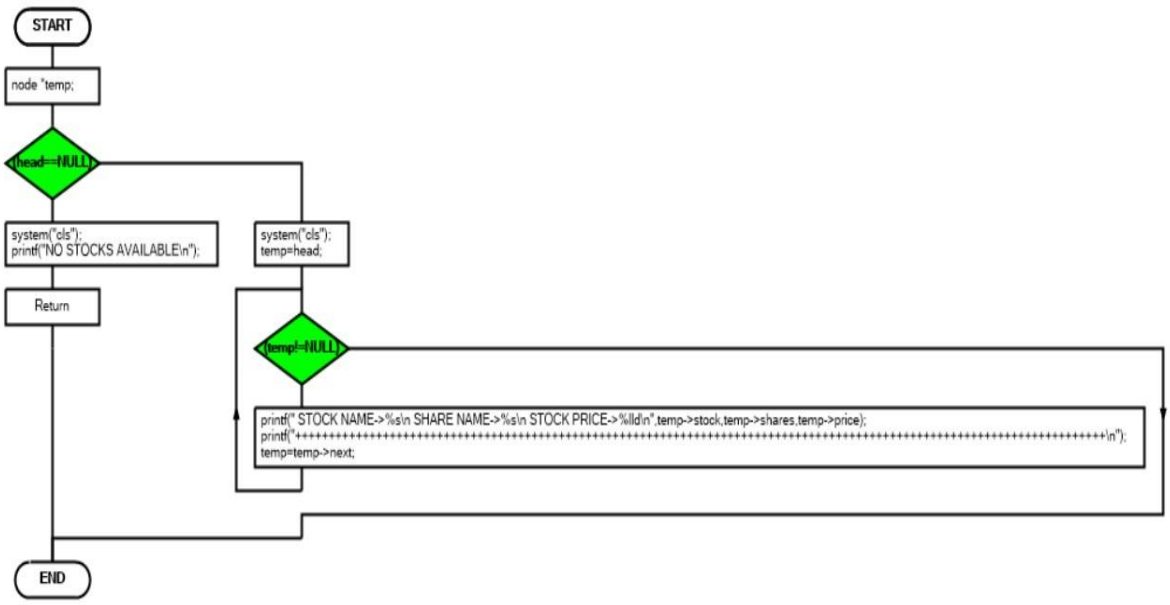
STEP 5:-display() function will display all the added stock details

FLOWCHART


START
int ch;
printf("\n++");
printf("\n");
printf("\t\t\t\t\t\tStock Table");
printf("\n");
printf("\n++");
printf("\n Enter >1< To Enter A New Stock.");
printf("\n Enter >2< Find price for a stock.");
printf("\n Enter >3< Display stocks.");
printf("\n Enter >0< To Exit.\n");
printf("\n Enter Your Choice : ");
scanf("%d",&ch);
(ch==1)
add_stock();
(ch==2)
find_stock();
(ch==3)
display_stock();
Break
(ch!=0)
END


START
node *nn,*temp;
nn=(node *)malloc(sizeof(node));
char s1[100],s2[100];
printf(" Enter Stock Name: ");
scanf("%s",s1);
strcpy(nn->stock,s1);
printf(" Enter Share Name: ");
scanf(" %s",s2);
strcpy(nn->shares,s2);
long long int p;
printf(" Enter Stock Price: ");
scanf("%lld",&p);
nn->price=p;
nn->next=NULL;
(head==NULL)
head=nn;
temp=head;
(temp->next!=NULL)
temp=temp->next;
temp->next=nn;
printf("New stock added\n");
system("cls");


START
printf("Enter the stock name: ");
char s1[100],s2[100];
scanf("%s",s1);
printf("Enter the stock share: ");
scanf(" %s",s2);
system("cls");
node *temp=head;
(temp!=NULL)
(strcmp(temp->stock,s1)==0
&& strcmp(temp->shares,s2)==0)
printf("Stock price is %lld\n",temp->price);
Return
temp=temp->next;
printf("NO STOCK AVAILABLE WITH THAT NAME/SHARE\n");
END


START
node *temp;
(head==NULL)
system("cls");
printf("NO STOCKS AVAILABLE\n");
Return
system("cls");
temp=head;
(temp!=NULL)
printf(" STOCK NAME->%s\n SHARE NAME->%s\n STOCK PRICE->%lld\n",temp->stock,temp->shares,temp->price);
printf("++\n");
temp=temp->next;
END

IMPLEMENTATION

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct node
{
        char stock[100];
        char shares[100];
        long long int price;
        struct node *next;
};
typedef struct node node;
node *head=NULL;
void add_stock()
{
        node *nn,*temp;
        nn=(node *)malloc(sizeof(node));
        char s1[100],s2[100];
        printf(" Enter Stock Name: ");
        scanf("%s",s1);
        strcpy(nn->stock,s1);
        printf(" Enter Stock Symbol: ");
        scanf(" %s",s2);
```

```
        strcpy(nn->shares,s2);
        long long int p;
        printf(" Enter Stock Price: ");
        scanf("%lld",&p);
        nn->price=p;
        nn->next=NULL;
        if(head==NULL)
        {
                head=nn;
        }
        else
        {
                temp=head;
                while(temp->next!=NULL)
                {
                        temp=temp->next;
                }
                temp->next=nn;
        }
        printf("New stock added\n");
        system("cls");
}
void find_stock()
{
```

```
        printf("Enter the stock name: ");
        char s1[100],s2[100];
        scanf("%s",s1);
        printf("Enter the stock Symbol: ");
        scanf(" %s",s2);
        system("cls");
        node *temp=head;
        while(temp!=NULL)
        {
                if(strcmp(temp->stock,s1)==0 && strcmp(temp->shares,s2)==0)
                {
                        printf("Stock price is %lld\n",temp->price);
                        return;
                }
                temp=temp->next;
        }
        printf("NO STOCK AVAILABLE WITH THAT NAME/SHARE\n");

}
void display_stock()
{
        node *temp;
        if(head==NULL)
```

```
	{
		system("cls");
		printf("NO STOCKS AVAILABLE\n");
		return;
	}
	else
	{
		system("cls");
		temp=head;
		while(temp!=NULL)
		{
			printf(" STOCK NAME->%s\n STOCK SYMBOL ->%s\n
STOCK PRICE->%lld\n",temp->stock,temp->shares,temp->price);

	printf("++++++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++\n");
			temp=temp->next;
		}

	}

}
int main()
{
```

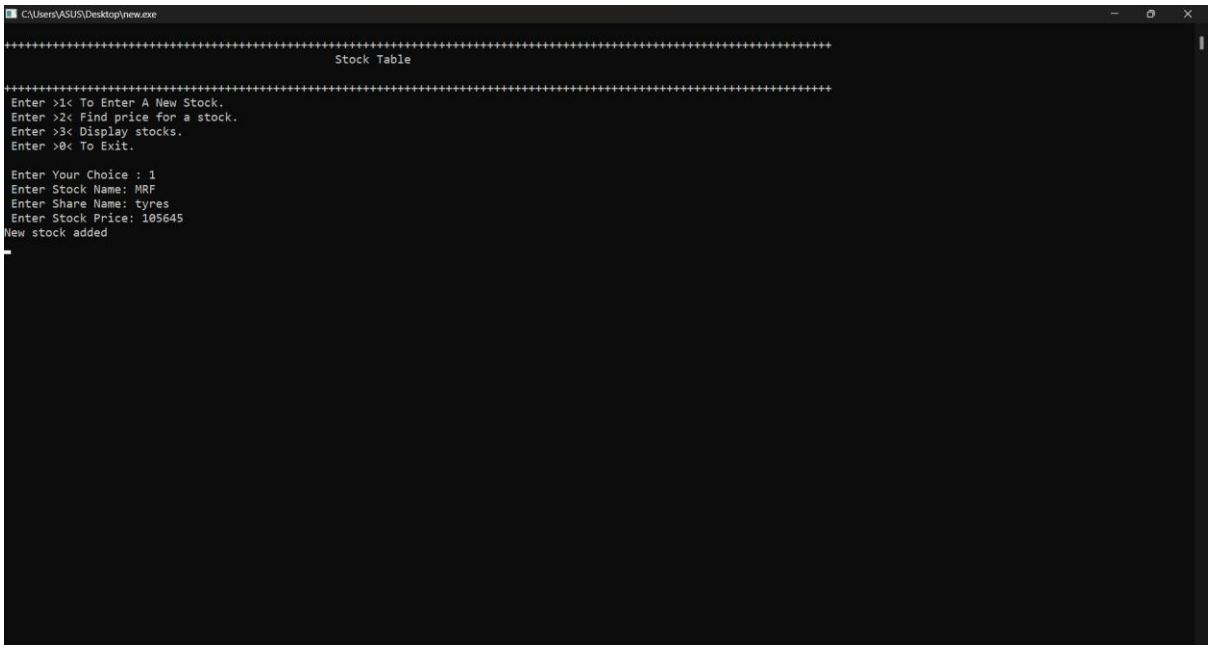
```
        int ch;

        do

        {printf("\n+++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++");

        printf("\n");

        printf("\t\t\t\t\t\tStock Table");

        printf("\n");

        printf("\n+++++++++++++++++++++++++++++++++++++++++++++
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++++++++++++++");

        printf("\n Enter >1< To Enter A New Stock.");

        printf("\n Enter >2< Find price for a stock.");

        printf("\n Enter >3< Display stocks.");

        printf("\n Enter >0< To Exit.\n");

        printf("\n Enter Your Choice : ");

                scanf("%d",&ch);

                if(ch==1)

                {

                        add_stock();

                }

                else if(ch==2)

                {

                        find_stock();

                }
```

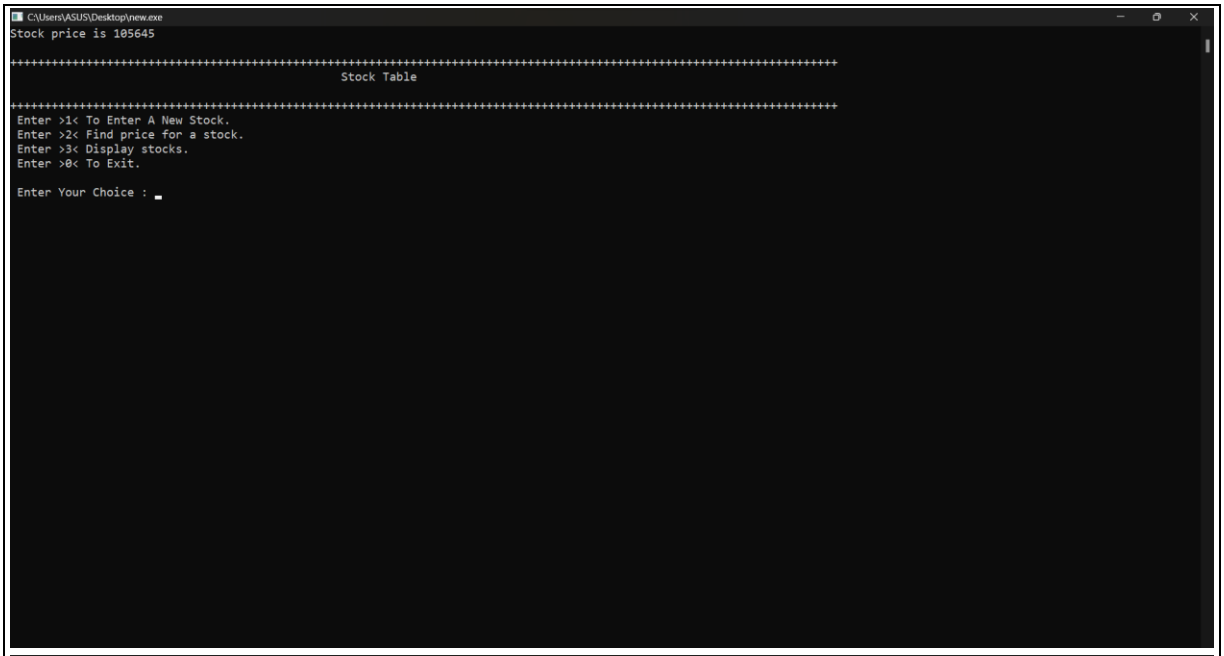
```
            else if(ch==3)
            {
                display_stock();
            }
            else
            {
                break;
            }
        }while(ch!=0);
}
```

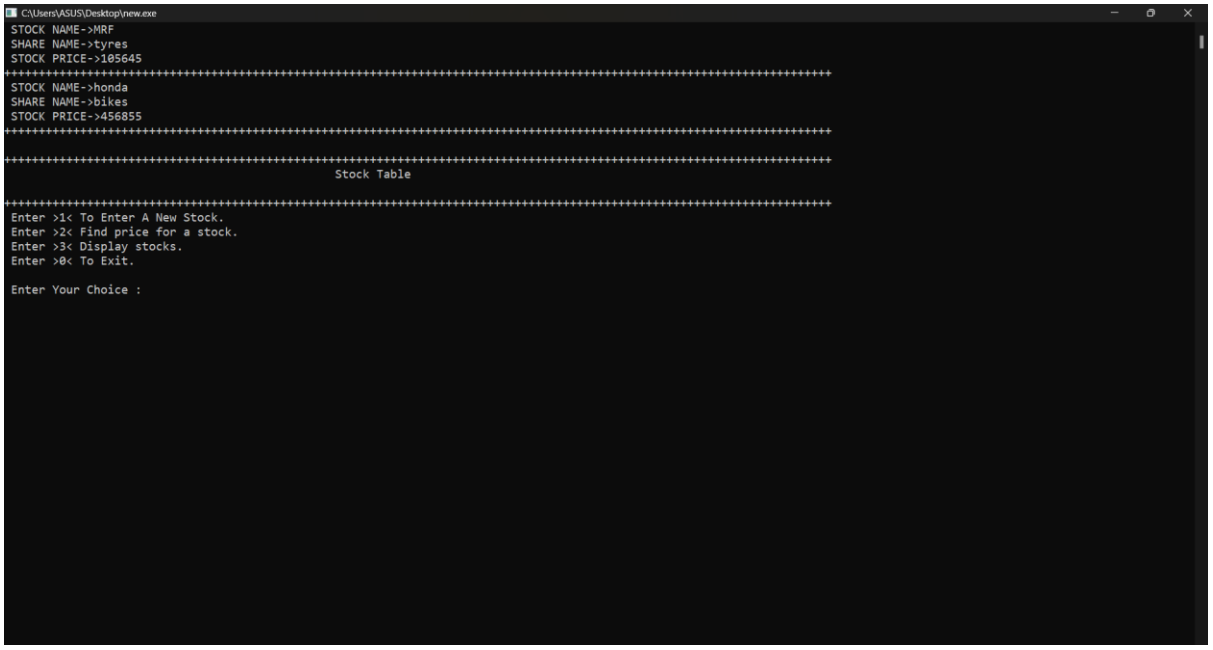
RESULTS AND SCREENSHOTS

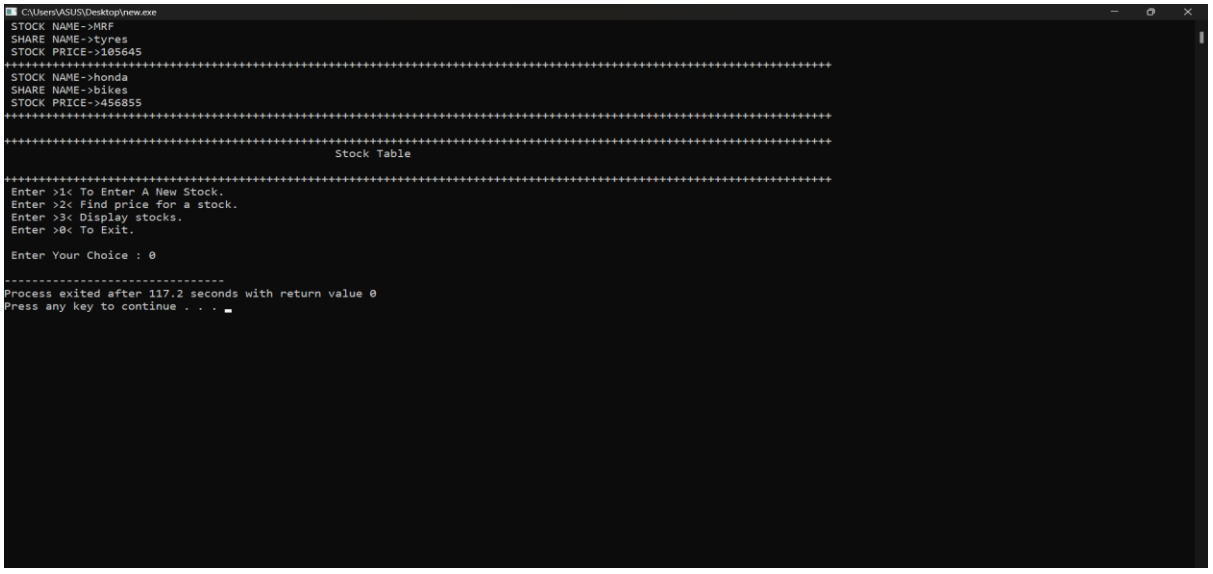
OUTPUTS

Screen Shots:




C:\Users\ASUS\Desktop\new.exe
Stock price is 105645
Stock Table
Enter >1< To Enter A New Stock.
Enter >2< Find price for a stock.
Enter >3< Display stocks.
Enter >0< To Exit.
Enter Your Choice :


C:\Users\ASUS\Desktop\new.exe
STOCK NAME->MRF
SHARE NAME->tyres
STOCK PRICE->105645
STOCK NAME->honda
SHARE NAME->bikes
STOCK PRICE->456855
Stock Table
Enter >1< To Enter A New Stock.
Enter >2< Find price for a stock.
Enter >3< Display stocks.
Enter >0< To Exit.
Enter Your Choice :


C:\Users\ASUS\Desktop\new.exe
STOCK NAME->MRF
SHARE NAME->tyres
STOCK PRICE->105645
STOCK NAME->honda
SHARE NAME->bikes
STOCK PRICE->456855
Stock Table
Enter >1< To Enter A New Stock.
Enter >2< Find price for a stock.
Enter >3< Display stocks.
Enter >0< To Exit.
Enter Your Choice : 0
Process exited after 117.2 seconds with return value 0
Press any key to continue . . .

CONCLUSION

We have successfully completed our project by using stacks concept and different data structures and datatypes like strings, characters, integers and some operators like increment and decrement operators and loops like ‘for’ loop and ‘while’ loop in our project. we understood the implementation of loops. we even used strings concept , string functions , stack implementation and about operation that we can do in stacks like push, pop, display in our project and we understood the implementation of string functions.we learnt what is Stock Table? And where it is used? And we had learnt the adavantages and disadvanatages of our project.The main thing is we understood about team work.