Stock Management System - Project File

1. Title Page

Project: Stock Management System

Course: Programming in C

Student: Kartik Gupta

SAP ID: 590025981

Instructor: Dr. Tanu Singh

University: University of Petroleum and Energy Studies

Date: December 2025

2. Abstract

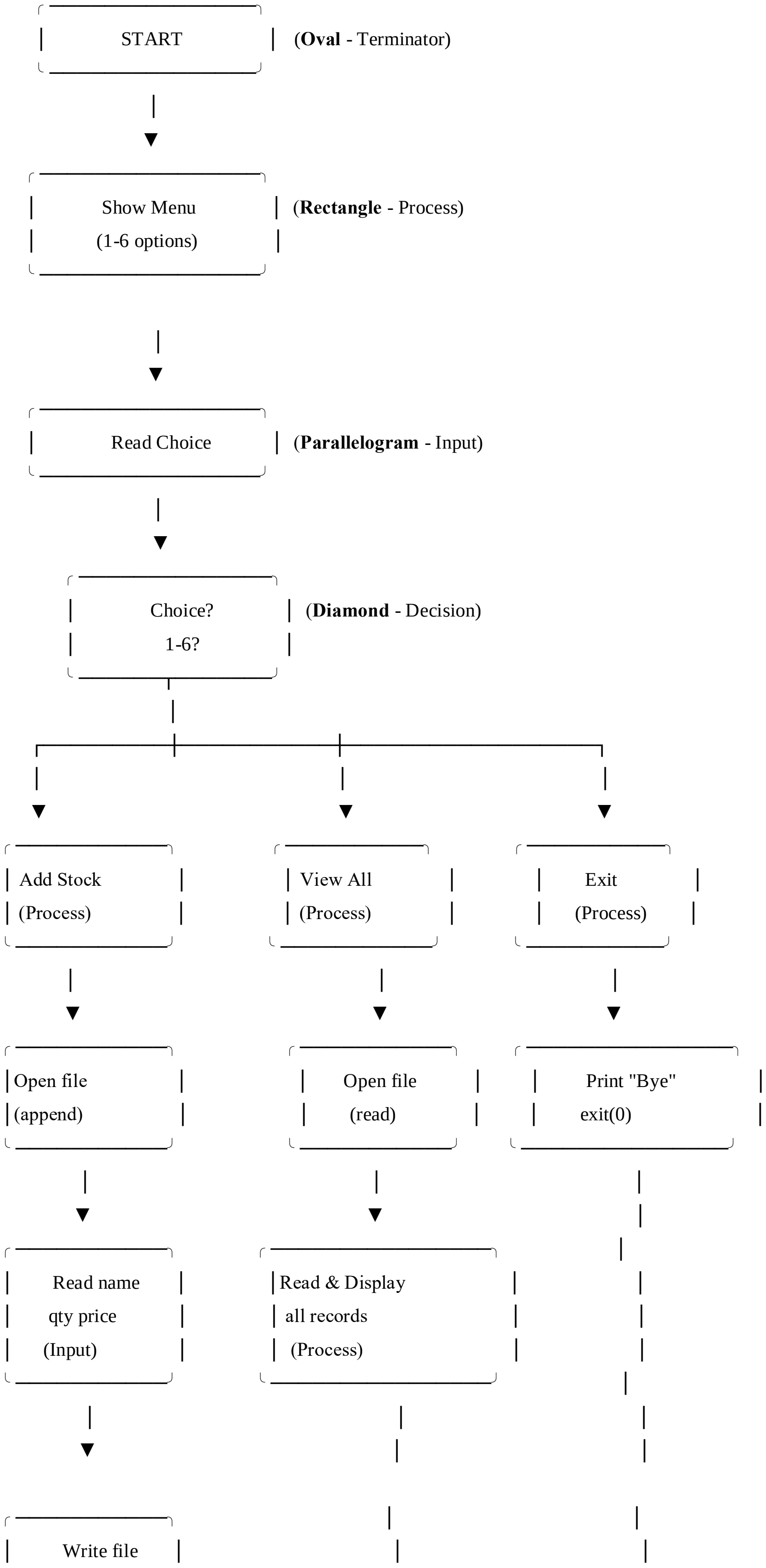
This project is a simple stock management system written in C. It manages stock items by allowing the user to add, view, search, update quantity, and delete stock records. Each stock item stores its name, quantity, and purchase price. The program uses file handling to save and retrieve data. The project demonstrates basic concepts of C programming including structures, file I/O, and string handling.

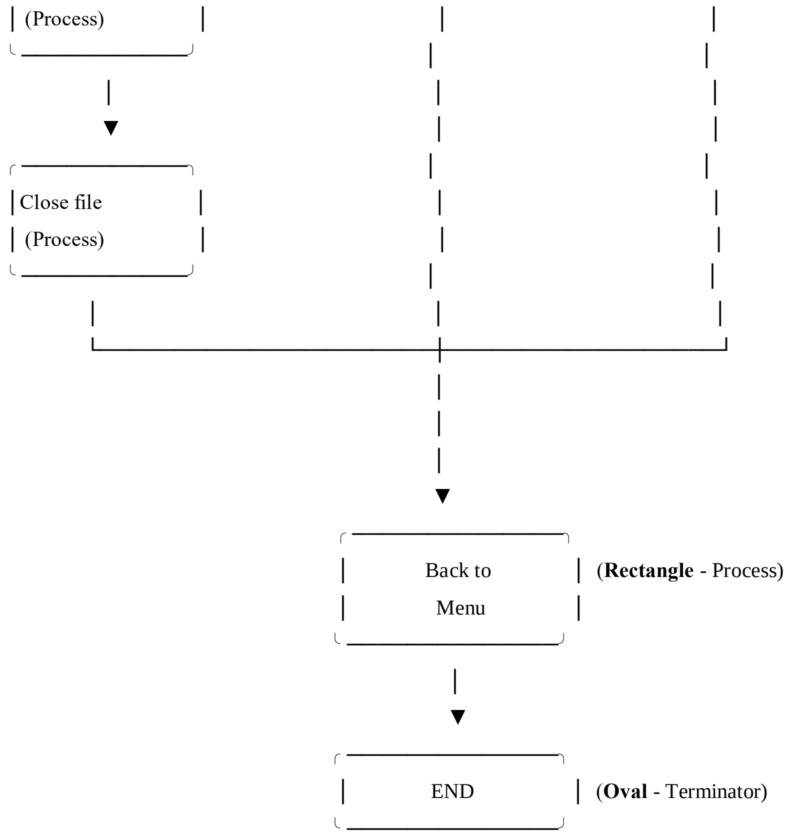
3. Problem Definition

Managing stocks manually can be difficult and prone to mistakes. This system helps automate stock records management easily using a computer program. It keeps track of stock names, their quantities, and purchase prices, making it easier for a user to organize and update stock data accurately.

4. System Design

Flowchart





Algorithm

1. Start
2. Display the main menu.
3. Read the user’s choice.
4. If choice = 1 (Add stock):
    - Open file in append mode.
    - Read stock name, quantity, and price from user.
    - Write these values to the file.
    - Close the file.
    - Go back to step 2.
5. If choice = 2 (View all stocks):
    - Open file in read mode.
    - If file not found, show “No stock data”.
    - Otherwise, read each record from file and print name, quantity, price.
    - Close the file.
    - Go back to step 2.
6. If choice = 3 (Search stock by name):
    - Open file in read mode.

- Read the name to search from user.
- For each record in file, compare name with search name.
- If match found, display that stock and set flag = found.
- After loop, if no match, print “Stock not found”.
- Close file.
- Go back to step 2.

7. If choice = 4 (Update stock quantity):
    - Open main file in read mode.
    - Open a temporary file in write mode.
    - Read the name to update and new quantity.
    - For each record in main file:
    - If name matches, change quantity to new value and set flag = found.
    - Write record (updated or same) into temp file.
    - Close both files.
    - Replace main file with temp file.
    - If flag = found, print “Updated”, else print “Stock not found”.
    - Go back to step 2.
8. If choice = 5 (Delete stock by name):
    - Open main file in read mode.
    - Open a temporary file in write mode.
    - Read the name to delete.
    - For each record in main file:
    - If name is not equal to delete name, write it to temp file.
    - If equal, skip writing and set flag = found.
    - Close both files.
    - Replace main file with temp file.
    - If flag = found, print “Deleted”, else print “Stock not found”.
    - Go back to step 2.
9. If choice = 6 (Exit):
    - Print exit message.
    - Go to step 10.
10. Stop

5. Implementation Details

- Used C language with basic commands only
- Used struct to define a stock item
- Used file stock_data.txt to store stock data persistently
- Wrote separate functions for add, view, search, update quantity, delete
- Used simple string handling with fgets for stock names allowing spaces
- Used loops, conditionals, and file handling functions

(fopen , fclose , fprintf , fscanf)

6. Testing Results

- Tested adding stocks with names having spaces
- Verified viewing all stocks lists all entries correctly
- Tested searching stocks by name with correct and wrong names
- Updated quantity of existing stocks successfully
- Deleted stocks by name and verified removal
- Program runs without crashing and handles invalid inputs gracefully

7. Conclusion and Future Work

The project successfully implements a basic stock management system in C. It is easy to extend for more features like sorting, multiple stock attributes, and graphical user interface. Future work can include improving user input validation and adding report generation.

8. References

- University of Petroleum and Energy Studies programming guide
- C Programming tutorials and references for file handling and struct usage

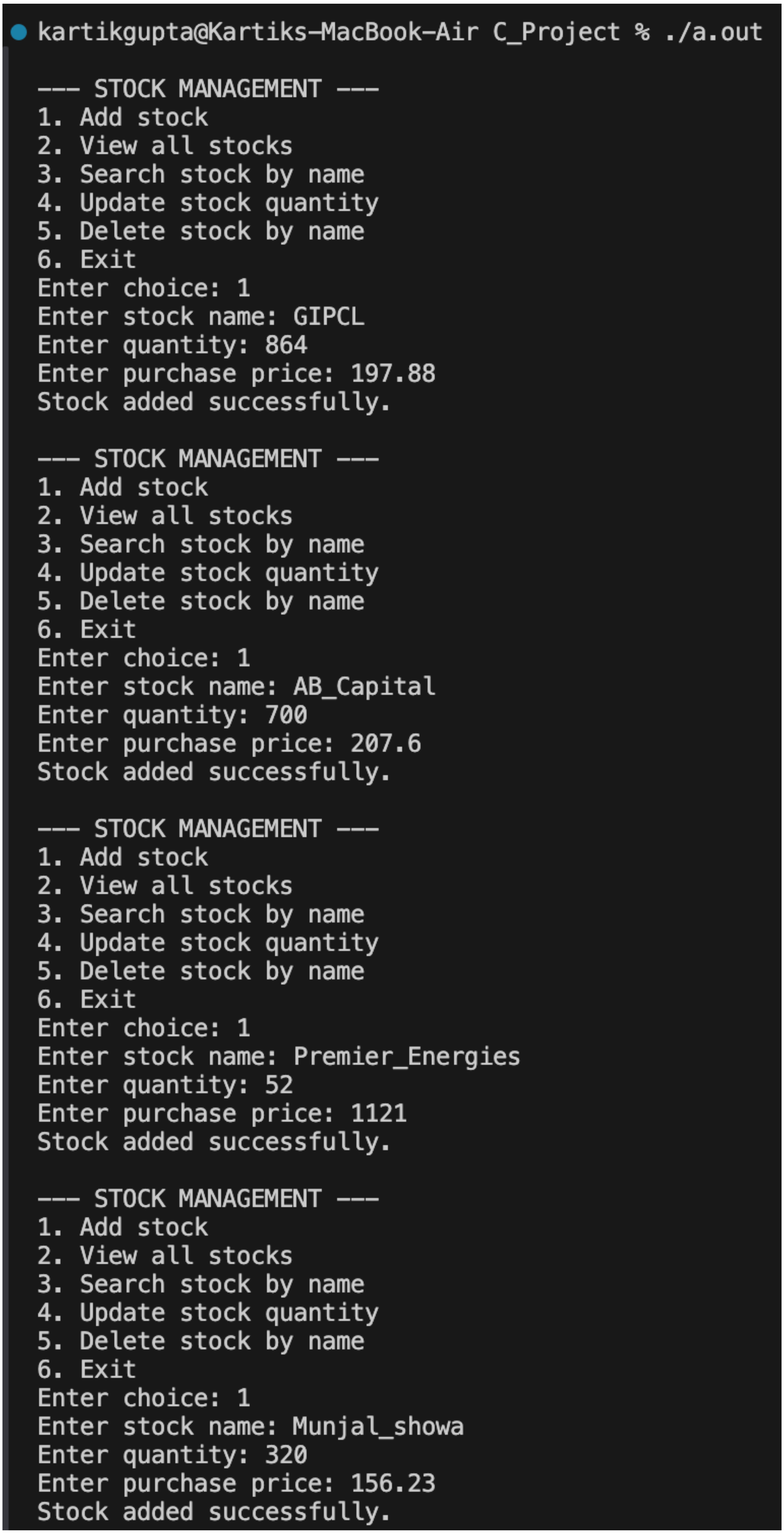
9. Appendix

- Source code files (stock.c , stock.h)
- Sample input/output screenshots (optional)

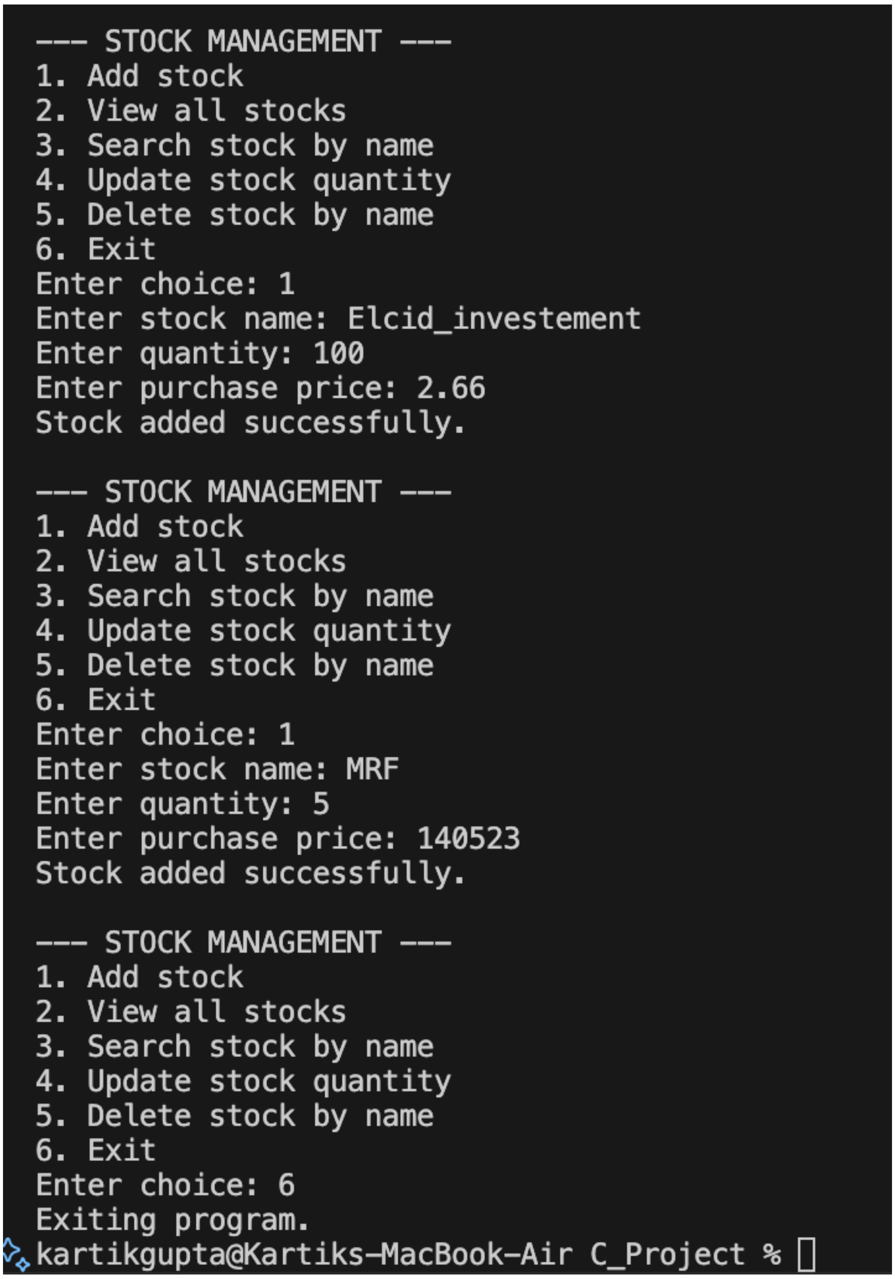
```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <string.h>
#include "../include/stock.h"

#define DATA_FILE "stock_data.txt"

void showMenu() {
    printf("\n--- STOCK MANAGEMENT ---\n");
    printf("1. Add stock\n");
    printf("2. View all stocks\n");
    printf("3. Search stock by name\n");
    printf("4. Update stock quantity\n");
    printf("5. Delete stock by name\n");
    printf("6. Exit\n");
    printf("Enter choice: ");
}

void addStock() {
    FILE *fp = fopen(DATA_FILE, "a");
    if (!fp) {
        printf("Error opening file.\n");
        return;
    }

    StockItem item;

    printf("Enter stock name: ");
    scanf("%49s", item.name);
    printf("Enter quantity: ");
    scanf("%d", &item.quantity);
    printf("Enter purchase price: ");
    scanf("%f", &item.price);

    fprintf(fp, "%s %d %.2f\n", item.name, item.quantity, item.price);
    fclose(fp);
    printf("Stock added successfully.\n");
}

void viewStock() {
    FILE *fp = fopen(DATA_FILE, "r");
    if (!fp) {
        printf("No stock data found.\n");
        return;
    }

    StockItem item;
    printf("\nName\tQuantity\tPrice\n");
    printf("----------------------------\n");

    while (fscanf(fp, "%49s %d %f", item.name, &item.quantity, &item.price) == 3) {
        printf("%s\t%d\t\t%.2f\n", item.name, item.quantity, item.price);
    }

    fclose(fp);
}

void searchStock() {
    FILE *fp = fopen(DATA_FILE, "r");
    if (!fp) {
        printf("No stock data found.\n");
        return;
    }

    char searchName[NAME_LEN];
    printf("Enter stock name to search: ");
    scanf("%49s", searchName);
    StockItem item;
    int found = 0;

    while (fscanf(fp, "%49s %d %f", item.name, &item.quantity, &item.price) == 3) {
        if (strcmp(item.name, searchName) == 0) {
            printf("\nStock found:\nName: %s\nQuantity: %d\nPrice: %.2f\n", item.name, item.quantity, item.price);
            found = 1;
            break;
        }
    }

    if (!found) {
        printf("Stock not found.\n");
    }

    fclose(fp);
}

void updateStock() {
    FILE *fp = fopen(DATA_FILE, "r");
    if (!fp) {
        printf("No stock data found.\n");
        return;
    }

    FILE *temp = fopen("temp.txt", "w");
    if (!temp) {
        printf("Error opening temporary file.\n");
        fclose(fp);
        return;
    }

    char searchName[NAME_LEN];
    printf("Enter stock name to update quantity: ");
    scanf("%49s", searchName);

    int newQuantity;
    printf("Enter new quantity: ");
    scanf("%d", &newQuantity);

    StockItem item;
    int found = 0;

    while (fscanf(fp, "%49s %d %f", item.name, &item.quantity, &item.price) == 3) {
        if (strcmp(item.name, searchName) == 0) {
            item.quantity = newQuantity;
            found = 1;
        }
        fprintf(temp, "%s %d %.2f\n", item.name, item.quantity, item.price);
    }

    fclose(fp);
    fclose(temp);

    remove(DATA_FILE);
    rename("temp.txt", DATA_FILE);

    if (found)
        printf("Stock quantity updated.\n");
    else
        printf("Stock not found.\n");
}

void deleteStock() {
    FILE *fp = fopen(DATA_FILE, "r");
    if (!fp) {
        printf("No stock data found.\n");
        return;
    }

    FILE *temp = fopen("temp.txt", "w");
    if (!temp) {
        printf("Error opening temporary file.\n");
        fclose(fp);
        return;
    }

    char deleteName[NAME_LEN];
    printf("Enter stock name to delete: ");
    scanf("%49s", deleteName);

    StockItem item;
    int found = 0;

    while (fscanf(fp, "%49s %d %f", item.name, &item.quantity, &item.price) == 3) {
        if (strcmp(item.name, deleteName) != 0) {
            fprintf(temp, "%s %d %.2f\n", item.name, item.quantity, item.price);
        } else {
            found = 1;
        }
    }

    fclose(fp);
    fclose(temp);

    remove(DATA_FILE);
    rename("temp.txt", DATA_FILE);

    if (found)
        printf("Stock deleted successfully.\n");
    else
```

```
        printf("Stock not found.\n");
}

int main() {
    int choice;

    while (1) {
        showMenu();
        if (scanf("%d", &choice) != 1) {
            printf("Invalid input.\n");
            break;
        }

        switch(choice) {
            case 1: addStock(); break;
            case 2: viewStock(); break;
            case 3: searchStock(); break;
            case 4: updateStock(); break;
            case 5: deleteStock(); break;
            case 6: printf("Exiting program.\n"); exit(0);
            default: printf("Invalid choice.\n");
        }
    }
    return 0;
}
```

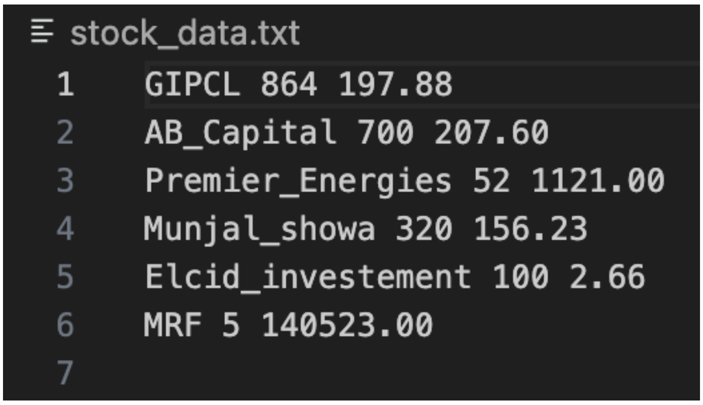


```
kartikgupta@Kartiks-MacBook-Air C_Project % ./a.out

--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 1
Enter stock name: GIPCL
Enter quantity: 864
Enter purchase price: 197.88
Stock added successfully.

--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 1
Enter stock name: AB_Capital
Enter quantity: 700
Enter purchase price: 207.6
Stock added successfully.

--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 1
Enter stock name: Premier_Energies
Enter quantity: 52
Enter purchase price: 1121
Stock added successfully.

--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 1
Enter stock name: Munjal_showa
Enter quantity: 320
Enter purchase price: 156.23
Stock added successfully.
```



```
--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 1
Enter stock name: Elcid_investement
Enter quantity: 100
Enter purchase price: 2.66
Stock added successfully.

--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 1
Enter stock name: MRF
Enter quantity: 5
Enter purchase price: 140523
Stock added successfully.

--- STOCK MANAGEMENT ---
1. Add stock
2. View all stocks
3. Search stock by name
4. Update stock quantity
5. Delete stock by name
6. Exit
Enter choice: 6
Exiting program.
kartikgupta@Kartiks-MacBook-Air C_Project %
```

Stock_data.txt file



```
GIPCL 864 197.88
AB_Capital 700 207.60
Premier_Energies 52 1121.00
Munjal_showa 320 156.23
Elcid_investement 100 2.66
MRF 5 140523.00

```