PROJECT REPORT

Course Code: CSEG1032

Course Title: Programming in C

Project Title: Stock portfolio Manager

Student Name: Bhoovi Verma

Global Id: 590025848

Semester: 1st

ABSTRACT

The Stock Portfolio Manager is a menu-driven C program that helps users manage their stock investments. It allows buying stocks from a predefined list, storing purchase prices, updating current market prices, and automatically calculating profit or loss. The program uses arrays, functions, macros, and string handling to organize portfolio data and improve readability. Color-coded output enhances clarity while viewing performance. Users can also sell stocks, after which the portfolio is updated by shifting entries. Overall, the project demonstrates practical use of core C concepts while providing a simple and effective stock management system.

OBJECTIVE

1. To develop a C program that allows users to efficiently manage a stock portfolio.
2. To implement features for buying, viewing, updating, and selling stocks using arrays and functions.
3. To automatically calculate profit or loss based on purchase and current market prices.
4. To apply concepts like macros, string handling, conditional statements, and formatted output in a real-world application.
5. To enhance user interaction with a menu-driven interface and color-coded output.
6. To demonstrate proper input validation, error handling, and data organization in C programming.

PROBLEM DEFINITION

Managing multiple stock investments manually becomes difficult as users must track purchase prices, current market values, and calculate profit or loss for each stock. There is no simple console-based system that allows beginners to buy, view, update, and sell stocks while automatically computing their portfolio performance.

This project aims to solve this problem by developing a C program that stores stock details, updates prices, calculates profit/loss, and provides an easy menu-driven interface to manage a small stock portfolio.

SYSTEM DESIGN AND ALGORITHM

SYSTEM DESIGN

Main Menu: Displays menu and manages user choices.

- **Buy:** Shows available stocks and saves selected stock with purchase price.
- **Portfolio:** Updates market prices, recalculates profit/loss, and displays the table.
- **Sell:** Deletes selected stock and shifts remaining data.
- **P/L:** Computes profit or loss for each stock.

ALGORITHM

Main Algorithm

1. Show menu.
2. Take user choice.
3. Call Buy / View / Sell functions.
4. Repeat until Exit.

Buy Stock

1. Show stock list.
2. Take choice and purchase price.
3. Store details and compute P/L.

Calculate P/L

P/L = current_price – purchase_price

View Portfolio

1. Update prices.
2. Recalculate P/L.

3. Display table and totals.

Sell Stock

1. Select stock.
2. Confirm.
3. Remove entry and shift array.

MAJOR FUNCTIONS

1. **display_menu()**

Displays the main menu and available operations for the user.

2. **buy_stock_logic()**

Allows the user to purchase a stock, enter purchase price, and save the details in the portfolio.

3. **calculate_profit_loss(int index)**

Computes profit or loss for a specific stock using its purchase and current price.

4. **update_current_prices()**

Takes updated market prices from the user and recalculates profit/loss for all stocks.

5. **display_portfolio_logic()**

Shows the complete portfolio in a formatted table with color-coded profit/loss and totals.

6. **sell_stock_logic()**

Enables selling of a selected stock, displays realized P/L, and removes the entry from the portfolio.

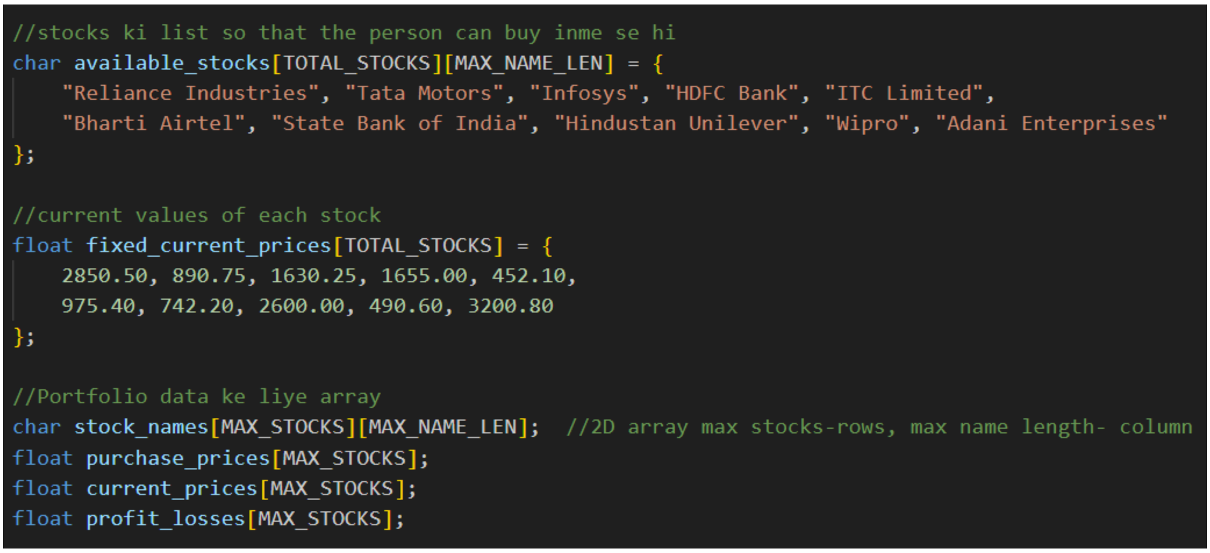
IMPLEMENTATION DETAILS

The Stock Portfolio Manager is implemented in C using arrays, structures, functions, and conditional logic.
The program maintains a fixed-size portfolio and allows the user to buy, view, and sell stocks using a menu-driven interface.

1. Data Structures Used

Arrays

Used to store stock details:



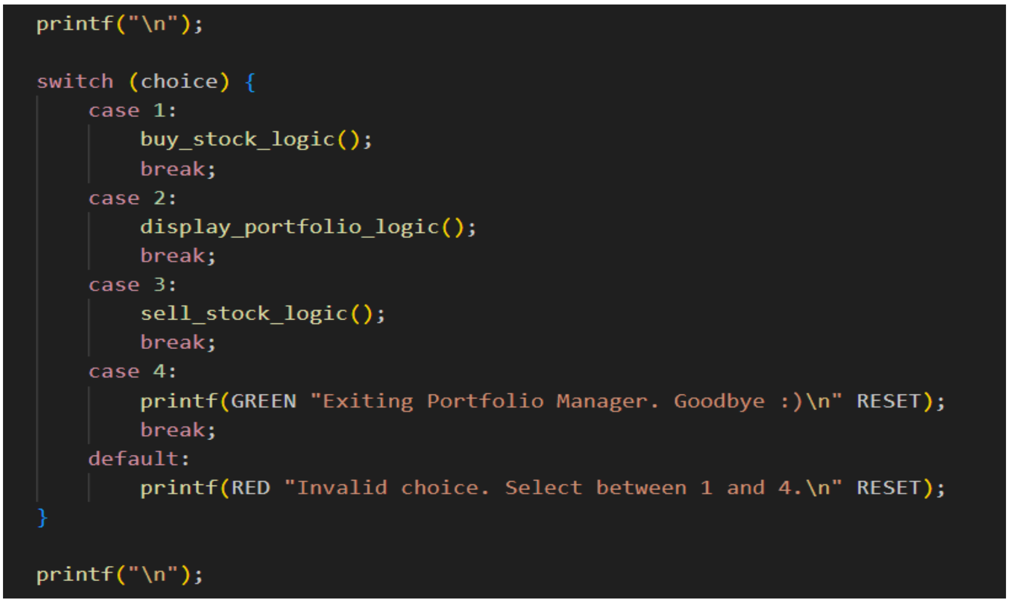
```c
//stocks ki list so that the person can buy inme se hi
char available_stocks[TOTAL_STOCKS][MAX_NAME_LEN] = {
    "Reliance Industries", "Tata Motors", "Infosys", "HDFC Bank", "ITC Limited",
    "Bharti Airtel", "State Bank of India", "Hindustan Unilever", "Wipro", "Adani Enterprises"
};

//current values of each stock
float fixed_current_prices[TOTAL_STOCKS] = {
    2850.50, 890.75, 1630.25, 1655.00, 452.10,
    975.40, 742.20, 2600.00, 490.60, 3200.80
};

//Portfolio data ke liye array
char stock_names[MAX_STOCKS][MAX_NAME_LEN];  //2D array max stocks-rows, max name length- column
float purchase_prices[MAX_STOCKS];
float current_prices[MAX_STOCKS];
float profit_losses[MAX_STOCKS];
```

2. Menu-Driven Interface

The program runs inside an infinite loop until the user selects Exit.



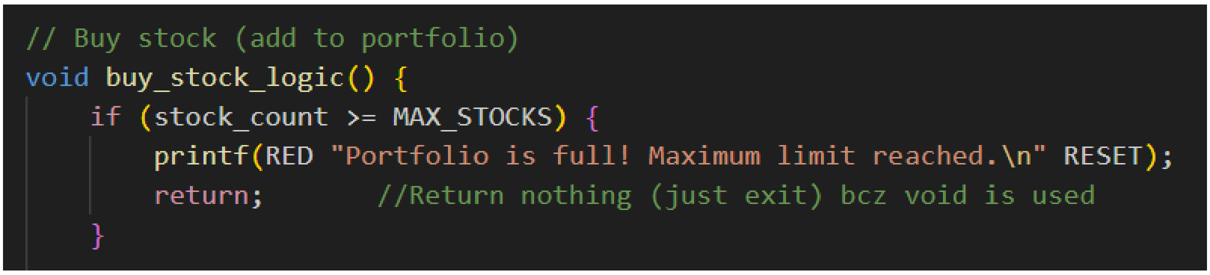
```c
printf("\n");

switch (choice) {
    case 1:
        buy_stock_logic();
        break;
    case 2:
        display_portfolio_logic();
        break;
    case 3:
        sell_stock_logic();
        break;
    case 4:
        printf(GREEN "Exiting Portfolio Manager. Goodbye :)\n" RESET);
        break;
    default:
        printf(RED "Invalid choice. Select between 1 and 4.\n" RESET);
}

printf("\n");
```

3. Buy Stock Function

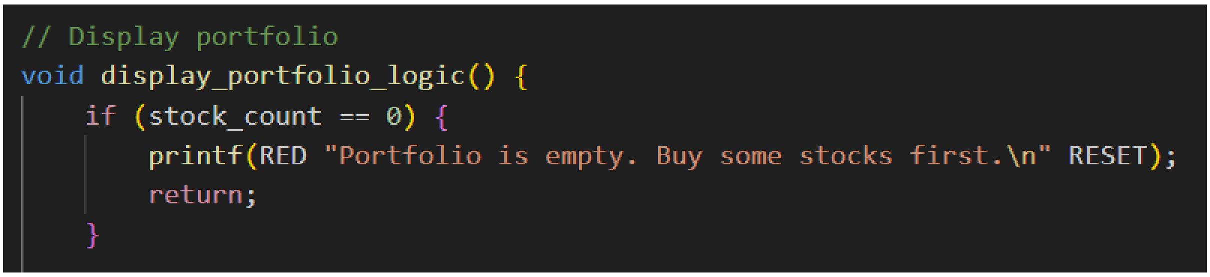
Stores stock name and purchase price in arrays.



```c
// Buy stock (add to portfolio)
void buy_stock_logic() {
    if (stock_count >= MAX_STOCKS) {
        printf(RED "Portfolio is full! Maximum limit reached.\n" RESET);
        return;       //Return nothing (just exit) bcz void is used
    }
```

4. Display Portfolio Function

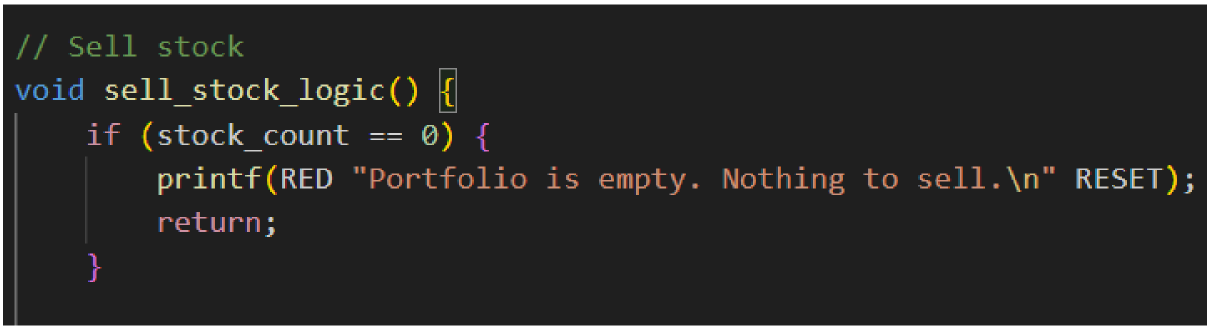
Calculates profit or loss for each stock and displays formatted output.



```c
// Display portfolio
void display_portfolio_logic() {
    if (stock_count == 0) {
        printf(RED "Portfolio is empty. Buy some stocks first.\n" RESET);
        return;
    }
```

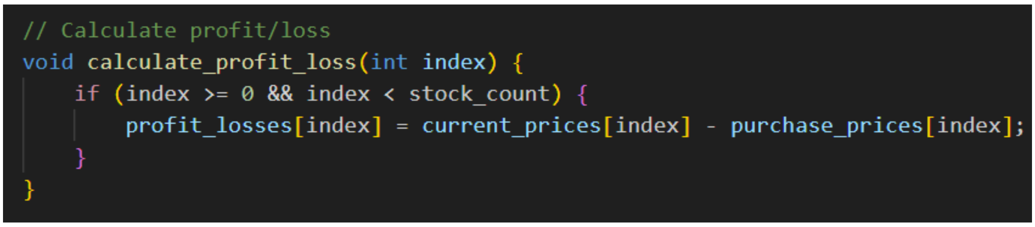
5. Sell Stock Function

Deletes a stock by shifting array elements.



```c
// Sell stock
void sell_stock_logic() {
    if (stock_count == 0) {
        printf(RED "Portfolio is empty. Nothing to sell.\n" RESET);
        return;
    }
```

6. Profit/Loss Calculation Logic



```c
// Calculate profit/loss
void calculate_profit_loss(int index) {
    if (index >= 0 && index < stock_count) {
        profit_losses[index] = current_prices[index] - purchase_prices[index];
    }
}
```

7. Testing & Results

The program was tested using different inputs to verify the correct functioning of stock purchase, display, and selling operations. Testing was performed on a GCC C compiler in a terminal environment.

SUMMARY OF RESULTS

- **All core features (Buy, View, Sell) work as expected.**
- **Profit/Loss calculations are accurate.**
- **Array shifting during selling works correctly.**
- **Input validation and error handling perform correctly.**
- **Output formatting with colors increases readability.**

Conclusion & Future Work

CONCLUSION

The Stock Portfolio Manager program successfully enables users to buy, view, and sell stocks while automatically calculating profit or loss. It uses arrays, functions, and a menu-driven interface to efficiently manage stock data. The system meets its objectives and provides a simple platform for understanding basic portfolio management using C.

FUTURE WORK

1. **Add file handling for saving portfolios.**
2. **Use dynamic memory for unlimited stocks.**
3. **Include sorting and filtering features.**
4. **Add GUI and charts for better visualization.**
5. **Integrate real-time price updates using APIs.**

References

GeeksforGeeks – Arrays, Functions, and File Handling in C.

Let us C book by Yashwant kanetkar