

~\OneDrive\Documents\Trading Shares Portfolio Management System\prog.c

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  #define MAX_STOCKS 100          // maximum number of stocks in portfolio
6  #define PREDEFINED_STOCKS 7    // number of predefined available stocks
7  #define PORTFOLIO_FILE "portfolio.txt" //file to save/load portfolio
8
9  //structure for predefine stock list
10 typedef struct {
11     char symbol[10];           // symbol for stock
12     float buy_price;           //buy price of stock
13     int available_quantity;    //availavble quantity of stock
14 } PredefinedStock;
15
16 //structure for user poortfolio
17 typedef struct {
18     char symbol[10];           //symbol of stock
19     int quantity;              //no.of stocks owned
20     float buy_price;           //buy price each stock
21     float current_price;       //current price
22 } Stock;
23
24 //array for predefined stocks
25 PredefinedStock predefined[] = {
26     {"AAPL", 150.00, 100},
27     {"GOOG", 2800.00, 50},
28     {"MSFT", 310.00, 150},
29     {"TSLA", 720.00, 80},
30     {"AMZN", 3400.00, 40},
31     {"NFLX", 450.00, 60},
32     {"SAMSUNG", 900.00, 70}
33 };
34
35 // Array to store user's stock portfolio
36 Stock portfolio[MAX_STOCKS];
37 int count = 0;          // Keeps track of how many stocks are in the user's portfolio
38
39 // Function declarations
40 void load_portfolio();
41 void save_portfolio();
42 void add_stock();
43 void remove_stock();
44 void view_portfolio();
45 void list_predefined_stocks();
46
47 int main() {
48     int choice;
```

```
49     load_portfolio();           // Load portfolio from file at program start
50
51     while (1) {
52         printf("\n--- Stock Portfolio Menu ---\n");
53         printf("1. Add Stock from List\n");
54         printf("2. Remove Stock\n");
55         printf("3. View Portfolio\n");
56         printf("4. Save\n");
57         printf("5. EXIT\n");
58         printf("Enter your choice: ");
59         scanf("%d", &choice);      //for reading user choice
60
61         switch (choice) {
62             case 1: add_stock(); break;
63             case 2: remove_stock(); break;
64             case 3: view_portfolio(); break;
65             case 4: save_portfolio(); break;
66             case 5: return 0;        // Exit program
67             default: printf("Invalid choice.\n");
68         }
69     }
70     return 0;
71 }
72
73 //for displaying predefined available stocks
74 void list_predefined_stocks() {
75     printf("\nAvailable Stocks:\n");
76     printf("%-3s %-10s %-15s %-10s\n", "No", "Symbol", "Buy Price", "Available");
77     for (int i = 0; i < PREDEFINED_STOCKS; i++) {
78         printf("%-3d %-10s %-15.2f %-10d\n",
79             i + 1, predefined[i].symbol, predefined[i].buy_price,
80             predefined[i].available_quantity);
81     }
82 }
83 // Add a stock from the predefined list to the portfolio
84 void add_stock() {
85     if (count >= MAX_STOCKS) {
86         printf("Portfolio full.\n");
87         return;
88     }
89
90     list_predefined_stocks();
91
92     int choice;
93     printf("Select stock (1-%d): ", PREDEFINED_STOCKS);
94     scanf("%d", &choice);
95
96     if (choice < 1 || choice > PREDEFINED_STOCKS) {
97         printf("Invalid selection.\n");
```

```
98     return;
99 }
100
101 int index = choice - 1;    // Convert to 0-based index
102
103 if (predefined[index].available_quantity <= 0) {    // No stock left to buy
104     printf("No available quantity left for this stock.\n");
105     return;
106 }
107
108 int qty;
109 printf("Enter quantity to buy (Available: %d): ", predefined[index].available_quantity);
110 scanf("%d", &qty);    // For Quantity user wants to buy
111
112 if (qty <= 0 || qty > predefined[index].available_quantity) {
113     printf("Invalid quantity.\n");
114     return;
115 }
116
117 Stock s;
118 strcpy(s.symbol, predefined[index].symbol);    // Copy stock symbol
119 s.buy_price = predefined[index].buy_price;    // Set buy price
120 s.quantity = qty;    // Set quantity
121
122 printf("Enter current market price: ");
123 scanf("%f", &s.current_price);    // Read current price from user
124
125 portfolio[count++] = s;    // Add stock to portfolio
126 predefined[index].available_quantity -= qty;
127
128 printf("Stock added to portfolio.\n");
129 }
130
131 // Remove a stock from the portfolio
132 void remove_stock() {
133     char symbol[10];
134     printf("Enter symbol to remove: ");
135     scanf("%s", symbol);
136
137     for (int i = 0; i < count; i++) {
138         if (strcmp(portfolio[i].symbol, symbol) == 0) {    // Match found
139             for (int j = 0; j < PREDEFINED_STOCKS; j++) {
140                 if (strcmp(predefined[j].symbol, symbol) == 0) {
141                     predefined[j].available_quantity += portfolio[i].quantity;
142                     break;
143                 }
144             }
145
146             for (int j = i; j < count - 1; j++) {
147                 portfolio[j] = portfolio[j + 1];
```

```
148         }
149         count--;
150         printf("Stock removed.\n");           // Symbol not found
151         return;
152     }
153 }
154 printf("Stock not found.\n");
155 }
156
157 // View all stocks in the portfolio
158 void view_portfolio() {
159     float total_profit = 0;
160
161     printf("\n%-10s %-10s %-10s %-10s %-10s\n", "Symbol", "Qty", "Buy", "Current", "P/L");
162     for (int i = 0; i < count; i++) {
163         float pl = (portfolio[i].current_price - portfolio[i].buy_price) *
portfolio[i].quantity;
164         total_profit += pl;
165
166         printf("%-10s %-10d %-10.2f %-10.2f %-10.2f\n",
167             portfolio[i].symbol,
168             portfolio[i].quantity,
169             portfolio[i].buy_price,
170             portfolio[i].current_price,
171             pl);
172     }
173
174     // Display overall profit/loss summary
175     printf("Total Profit/Loss: %.2f\n", total_profit);
176     if (total_profit > 0) {
177         printf("Overall Status: Profit = %.2f\n", total_profit);
178     } else if (total_profit < 0) {
179         printf("Overall Status: Loss = %.2f\n", -total_profit);
180     } else {
181         printf("Overall Status: No profit and no loss.\n");
182     }
183 }
184
185 // Load portfolio data from file
186 void load_portfolio() {
187     FILE *fp = fopen(PORTFOLIO_FILE, "r");
188     if (!fp) {
189         printf("No saved portfolio found.\n");
190         return;
191     }
192
193     fscanf(fp, "%d", &count);
194     for (int i = 0; i < count; i++) {
195         float dummy_pl; // to read and ignore stored profit/loss
196         fscanf(fp, "%s %d %f %f %f",
```

```
197         portfolio[i].symbol,
198         &portfolio[i].quantity,
199         &portfolio[i].buy_price,
200         &portfolio[i].current_price,
201         &dummy_pl);
202     }
203
204     fclose(fp);
205     printf("Portfolio loaded from file.\n");
206 }
207
208 // Save portfolio data to text file
209 void save_portfolio() {
210     FILE *fp = fopen(PORTFOLIO_FILE, "w");
211     if (!fp) {
212         perror("Error opening portfolio.txt");
213         return;
214     }
215
216     fprintf(fp, "%d\n", count);    // Write number of stocks
217     for (int i = 0; i < count; i++) {
218         float pl = (portfolio[i].current_price - portfolio[i].buy_price) *
portfolio[i].quantity;
219         fprintf(fp, "%s %d %.2f %.2f %.2f\n",
220             portfolio[i].symbol,
221             portfolio[i].quantity,
222             portfolio[i].buy_price,
223             portfolio[i].current_price,
224             pl);
225     }
226
227     fclose(fp);
228     printf("Portfolio saved to file with profit/loss.\n");
229 }
230
```