

Grocery Management System

1. Project Titel:

Grocery Management System.

2. Submitted By:

- Mir Ashraf Adib (242-35-248)
- Faysal Ahmed (242-35-576)
- Yacin Ahmed (242-35-575)

3. Introduction:

This is a grocery management system where there are two characters (Admin and Customer).

The goal of this project is to demonstrate the use of different fundamental concepts of programming such as data types, array, structure, conditional statements, loops, string and functions in an easy, real-life scenario.

4. project Features:

- Admin and customer can register and login in in the system.
- Admin creates product list, quantity of the product and the price of the product.
- Customer selects the product and adds to his cart and order's the selected item.
- After the work is done they can easily log out from the system.

5. Concept explained:

- **Structures (struct):**

We use struct to define User and Product data types.

```
typedef struct {  
    char username[50];  
    char password[50];  
} User;
```

Stores username and password for Admin & Customer.

```
typedef struct {  
    int code;  
    char name[50];  
    float price;  
    int quantity;  
} Product;
```

- **Global Variables:**

Store data globally for easy access.

User admin, customer; // Stores admin & customer info

Product products[100]; // Array to store up to 100 products

int productCount = 0; // Keeps track of the number of products

- **main() Function**

It starts the program and calls the mainMenu() function.

```
int main() {  
    mainMenu();  
    return 0;  
}
```

- **mainMenu() – Main Interface**

This function displays the menu for Admin and Customer options.

```
void mainMenu() {
    int choice;
    while (1) { // Infinite loop to keep showing menu
        printf("\n=== Grocery Management System ===\n");
        printf("1. Admin Sign Up\n2. Admin Login\n3. Customer Sign Up\n4. Customer Login\n5. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1: adminSignUp(); break;
            case 2: adminLogin(); break;
            case 3: customerSignUp(); break;
            case 4: customerLogin(); break;
            case 5: exit(0);
            default: printf("Invalid choice! Try again.\n");
        }
    }
}
```

- **Admin Functions**

adminSignUp() – Register Admin

```
void adminSignUp() {
    printf("\n=== Admin Sign Up ===\n");
    printf("Enter username: ");
    scanf("%s", admin.username);
    printf("Enter password: ");
    scanf("%s", admin.password);
    printf("Admin registered successfully!\n");
}
```

- **Admin Menu**

Purpose:

Provides Admin functionalities like Adding Products & Processing Orders.

```
void adminMenu() {  
    int choice;  
    while (1) {  
        printf("\n=== Admin Panel ===\n");  
        printf("1. Add Product\n2. Process Orders\n3. Logout\n");  
        printf("Enter choice: ");  
        scanf("%d", &choice);  
  
        switch (choice) {  
            case 1: addProduct(); break;  
            case 2: processOrders(); break;  
            case 3: logout(); return;  
            default: printf("Invalid choice! Try again.\n");  
        }  
    }  
}
```

- **addProduct() – Admin Adds Product**

```
void addProduct() {  
    printf("\n=== Add Product ===\n");  
    printf("Enter Product Code: ");  
    scanf("%d", &products[productCount].code);  
    printf("Enter Product Name: ");  
    scanf("%s", products[productCount].name);  
    printf("Enter Product Price: ");  
    scanf("%f", &products[productCount].price);  
    printf("Enter Product Quantity: ");  
    scanf("%d", &products[productCount].quantity);  
    productCount++;  
    printf("Product added successfully!\n");  
}
```

- **Customer Menu**

Purpose:

Provides **Customer functionalities** like **Viewing & Purchasing Products**.

```
void customerMenu() {  
    int choice;  
    while (1) {  
        printf("\n=== Customer Panel ===\n");  
        printf("1. View Products\n2. Purchase Product\n3. Logout\n");  
        printf("Enter choice: ");  
        scanf("%d", &choice);  
  
        switch (choice) {  
            case 1: viewProducts(); break;  
            case 2: purchaseProduct(); break;  
            case 3: logout(); return;  
            default: printf("Invalid choice! Try again.\n");  
        }  
    }  
}
```

- **viewProducts() – Show Available Products**

```
void viewProducts() {  
    printf("\n=== Available Products ===\n");  
    for (int i = 0; i < productCount; i++) {  
        printf("Code: %d | Name: %s | Price: %.2f | Quantity: %d\n",  
            products[i].code, products[i].name, products[i].price,  
            products[i].quantity);  
    }  
}
```

- **purchaseProduct() – Customer Buys Items**

```
void purchaseProduct() {
    int code, qty, found = 0;
    printf("\n=== Purchase Product ===\n");
    printf("Enter Product Code: ");
    scanf("%d", &code);
    printf("Enter Quantity: ");
    scanf("%d", &qty);

    for (int i = 0; i < productCount; i++) {
        if (products[i].code == code) {
            found = 1;
            if (products[i].quantity >= qty) {
                products[i].quantity -= qty;
                printf("Purchase successful!\n");
            } else {
                printf("Not enough stock available!\n");
            }
            break;
        }
    }
    if (!found) {
        printf("Product not found!\n");
    }
}
```

- **logout() – Logout and Return to Main Menu**

```
void logout() {
    printf("Logging out...\n");
    mainMenu();
}
```

6. Code:

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

// Structure for User
typedef struct {
    char username[50];
    char password[50];
} User;

// Structure for Product
typedef struct {
    int code;
    char name[50];
    float price;
    int quantity;
} Product;

// Global Variables
User admin, customer;
Product products[100];
int productCount = 0;

// Function Declarations
void mainMenu();
void adminMenu();
void customerMenu();
void adminSignUp();
void adminLogin();
void customerSignUp();
void customerLogin();
void addProduct();
```

```
void viewProducts();  
void purchaseProduct();  
void processOrders();  
void logout();
```

```
// Main Function
```

```
int main() {  
    mainMenu();  
    return 0;  
}
```

```
// Main Menu
```

```
void mainMenu() {  
    int choice;  
    while (1) {  
        printf("\n=== Grocery Management System ===\n");  
        printf("1. Admin Sign Up\n2. Admin Login\n3. Customer Sign Up\n4.  
Customer Login\n5. Exit\n");  
        printf("Enter choice: ");  
        scanf("%d", &choice);  
  
        switch (choice) {  
            case 1: adminSignUp(); break;  
            case 2: adminLogin(); break;  
            case 3: customerSignUp(); break;  
            case 4: customerLogin(); break;  
            case 5: exit(0);  
            default: printf("Invalid choice! Try again.\n");  
        }  
    }  
}
```

```
// Admin Sign Up
```

```
void adminSignUp() {
```



```
printf("\n=== Admin Sign Up ===\n");
printf("Enter username: ");
scanf("%s", admin.username);
printf("Enter password: ");
scanf("%s", admin.password);
printf("Admin registered successfully!\n");
}
```

// Admin Login

```
void adminLogin() {
    char username[50], password[50];
    printf("\n=== Admin Login ===\n");
    printf("Enter username: ");
    scanf("%s", username);
    printf("Enter password: ");
    scanf("%s", password);

    if (strcmp(username, admin.username) == 0 && strcmp(password,
admin.password) == 0) {
        printf("Login successful!\n");
        adminMenu();
    } else {
        printf("Invalid credentials! Try again.\n");
    }
}
```

// Admin Menu

```
void adminMenu() {
    int choice;
    while (1) {
        printf("\n=== Admin Panel ===\n");
        printf("1. Add Product\n2. Process Orders\n3. Logout\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
```

```
        switch (choice) {
            case 1: addProduct(); break;
            case 2: processOrders(); break;
            case 3: logout(); return;
            default: printf("Invalid choice! Try again.\n");
        }
    }
}
```

// Add Product

```
void addProduct() {
    printf("\n=== Add Product ===\n");
    printf("Enter Product Code: ");
    scanf("%d", &products[productCount].code);
    printf("Enter Product Name: ");
    scanf("%s", products[productCount].name);
    printf("Enter Product Price: ");
    scanf("%f", &products[productCount].price);
    printf("Enter Product Quantity: ");
    scanf("%d", &products[productCount].quantity);

    productCount++;
    printf("Product added successfully!\n");
}
```

// Process Orders (Placeholder)

```
void processOrders() {
    printf("\n=== Process Orders ===\n");
    printf("Feature coming soon...\n");
}
```

// Customer Sign Up

```
void customerSignUp() {
```

```

    printf("\n=== Customer Sign Up ===\n");
    printf("Enter username: ");
    scanf("%s", customer.username);
    printf("Enter password: ");
    scanf("%s", customer.password);
    printf("Customer registered successfully!\n");
}

// Customer Login
void customerLogin() {
    char username[50], password[50];
    printf("\n=== Customer Login ===\n");
    printf("Enter username: ");
    scanf("%s", username);
    printf("Enter password: ");
    scanf("%s", password);

    if (strcmp(username, customer.username) == 0 && strcmp(password,
customer.password) == 0) {
        printf("Login successful!\n");
        customerMenu();
    } else {
        printf("Invalid credentials! Try again.\n");
    }
}

// Customer Menu
void customerMenu() {
    int choice;
    while (1) {
        printf("\n=== Customer Panel ===\n");
        printf("1. View Products\n2. Purchase Product\n3. Logout\n");
        printf("Enter choice: ");
        scanf("%d", &choice);
    }
}

```

```

        switch (choice) {
            case 1: viewProducts(); break;
            case 2: purchaseProduct(); break;
            case 3: logout(); return;
            default: printf("Invalid choice! Try again.\n");
        }
    }
}

```

// View Products

```

void viewProducts() {
    printf("\n=== Available Products ===\n");
    for (int i = 0; i < productCount; i++) {
        printf("Code: %d | Name: %s | Price: %.2f | Quantity: %d\n",
            products[i].code, products[i].name, products[i].price,
            products[i].quantity);
    }
}

```

// Purchase Product

```

void purchaseProduct() {
    int code, qty, found = 0;
    printf("\n=== Purchase Product ===\n");
    printf("Enter Product Code: ");
    scanf("%d", &code);
    printf("Enter Quantity: ");
    scanf("%d", &qty);

    for (int i = 0; i < productCount; i++) {
        if (products[i].code == code) {
            found = 1;
            if (products[i].quantity >= qty) {
                products[i].quantity -= qty;
            }
        }
    }
}

```

```

        printf("Purchase successful!\n");
    } else {
        printf("Not enough stock available!\n");
    }
    break;
}
}
if (!found) {
    printf("Product not found!\n");
}
}

// Logout
void logout() {
    printf("Logging out...\n");
    mainMenu();
}

```

Conclusion: Through this project, I have successfully demonstrated the use of data types, array, structure, conditional statements, loops, string and functions. The project simulates a simple and helpful mental health support tool and represents how programming concepts can be applied in real-life scenarios. This project also taught me how to break down problems, plan functions, and use structures and arrays effectively.