Pet Grooming Salon Management System

Name: Sania Kunjumon Philip

Roll No: 66

Course Name: Programming In C

Date: 17/07/24

Introduction

Brief Overview of the Project

The Pet Grooming Salon Management System is designed to help manage appointments, client details, and grooming services for a pet grooming salon. It simplifies the process of scheduling appointments, tracking client information, and managing services offered to pets.

Problem Statement

Managing a pet grooming salon involves keeping track of numerous appointments, client details, and services provided. Manual management can lead to errors, missed appointments, and disorganization, which can negatively impact business operations and customer satisfaction.

Objective

The objective of this project is to develop a simple yet efficient system for managing pet grooming services, scheduling appointments, and maintaining client records. The system should be easy to use and help streamline the salon's operations.

System Requirements

Hardware Requirements

- A computer with a minimum of 2GB RAM.
- 100MB of free disk space.
- Monitor, keyboard, and mouse.

Software Requirements

- Operating System: Windows
- C Compiler (e.g., GCC).
- Text Editor or Integrated Development Environment (IDE) for writing and editing C code.

Design and Development

Description of the Program Logic

The program is designed using a simple menu-driven approach where users can add client details, display all clients, and schedule appointments. The data is stored in an array of structures, each containing client information.

Pseudocode

```
BEGIN
  Initialize clientCount to 0
  WHILE true
    DISPLAY menu options
    READ choice
    IF choice is 1
      CALL addClient function
    ELSE IF choice is 2
      CALL displayClients function
    ELSE IF choice is 3
      CALL scheduleAppointment function
    ELSE IF choice is 4
      EXIT program
    ELSE
      DISPLAY "Invalid choice"
  END WHILE
END


FUNCTION addClient
  PROMPT for client details
  READ and store client details in the next available array position
  INCREMENT clientCount
END FUNCTION
```

```
FUNCTION displayClients
  IF no clients
    DISPLAY "No clients to display"
  ELSE
    FOR each client in clients array
      DISPLAY client details
    END FOR
  END IF
END FUNCTION

FUNCTION scheduleAppointment
  PROMPT for client ID
  FIND client by ID
  IF client found
    PROMPT for service type and appointment date
    STORE service type and appointment date
    DISPLAY "Appointment scheduled"
  ELSE
    DISPLAY "Client ID not found"
  END IF
END FUNCTION
```

Testing and Results

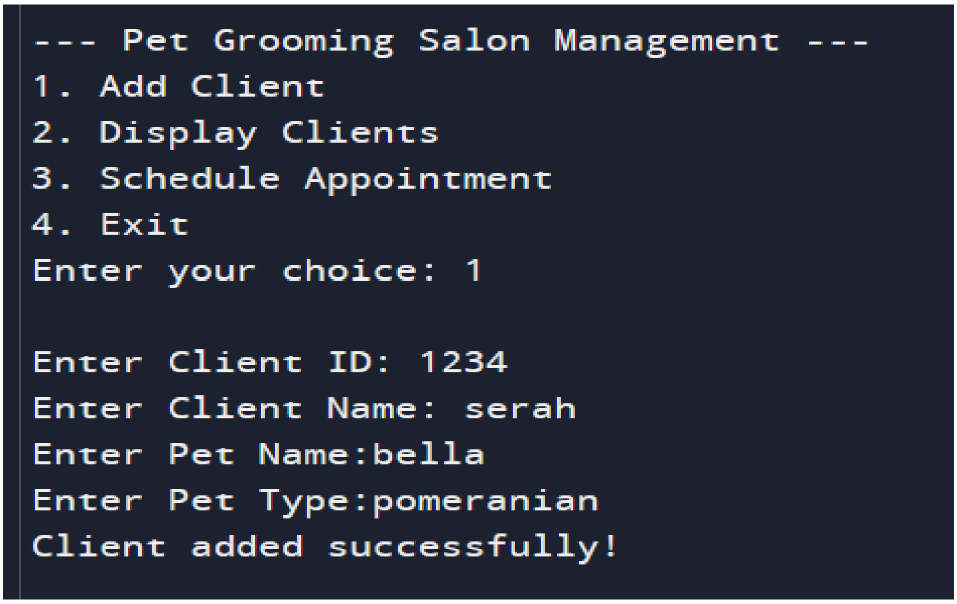
Test Cases

1. **Add Client:**
    - Input: Client ID: 1, Client Name: "John", Pet Name: "Buddy", Pet Type: "Dog".
    - Expected Result: Client added successfully.
2. **Display Clients:**
    - Input: Display Clients.
    - Expected Result: List of all clients with their details.
3. **Schedule Appointment:**
    - Input: Client ID: 1, Service Type: "Haircut", Appointment Date: "2024-07-16".
    - Expected Result: Appointment scheduled successfully for Client ID 1.

Output Screenshots or Results

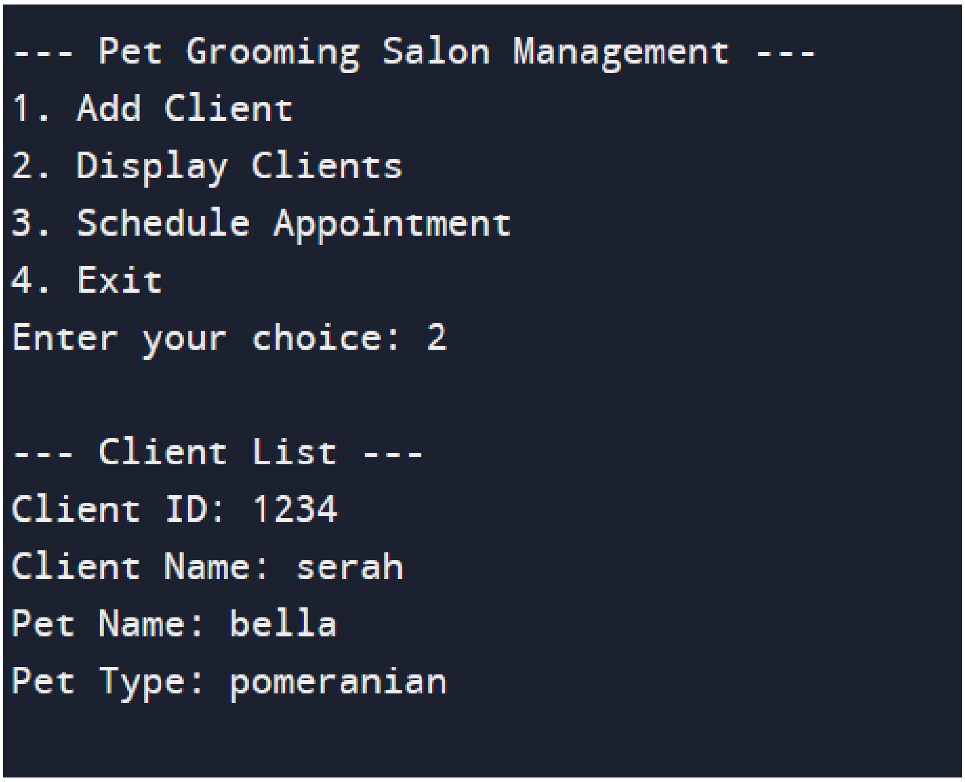
Here are some sample outputs based on the test cases above:

1.Add Client Output:



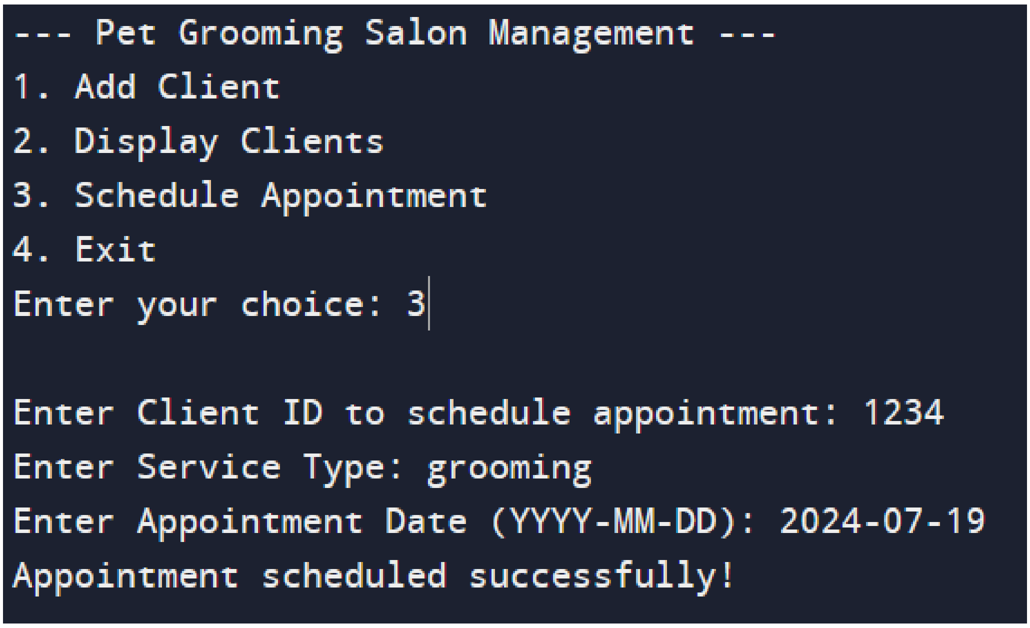
```
--- Pet Grooming Salon Management ---
1. Add Client
2. Display Clients
3. Schedule Appointment
4. Exit
Enter your choice: 1

Enter Client ID: 1234
Enter Client Name: serah
Enter Pet Name:bella
Enter Pet Type:pomeranian
Client added successfully!
```

2.Display Clients Output:



```
--- Pet Grooming Salon Management ---
1. Add Client
2. Display Clients
3. Schedule Appointment
4. Exit
Enter your choice: 2

--- Client List ---
Client ID: 1234
Client Name: serah
Pet Name: bella
Pet Type: pomeranian
```

3.Schedule Appointment Output:



```
--- Pet Grooming Salon Management ---
1. Add Client
2. Display Clients
3. Schedule Appointment
4. Exit
Enter your choice: 3

Enter Client ID to schedule appointment: 1234
Enter Service Type: grooming
Enter Appointment Date (YYYY-MM-DD): 2024-07-19
Appointment scheduled successfully!
```

Discussion of Results

The test cases demonstrated that the system can successfully add clients, display client information, and schedule appointments. The outputs matched the expected results, indicating that the system functions as intended.

Conclusion

Summary of the Project

The Pet Grooming Salon Management System provides an efficient way to manage client details, schedule appointments, and track grooming services. It simplifies salon operations, reduces the likelihood of errors, and improves overall customer service.

Future Enhancements

Future enhancements could include:

- Adding functionality to update and delete client details.
- Implementing a database to store client information.
- Developing a graphical user interface (GUI) for ease of use.

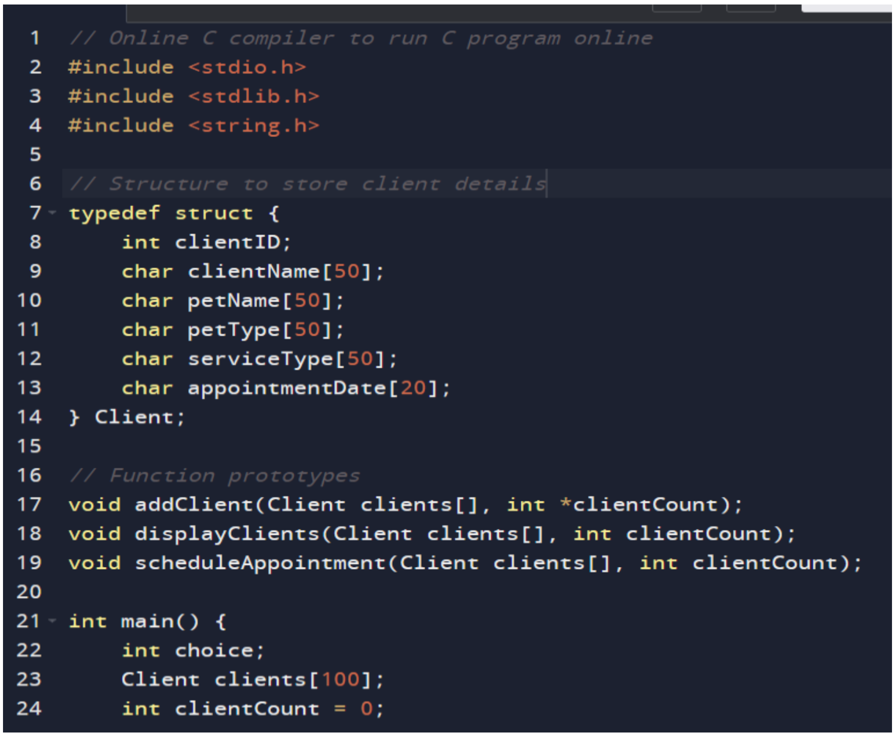
- Adding features for sending appointment reminders via email or SMS.

Reference

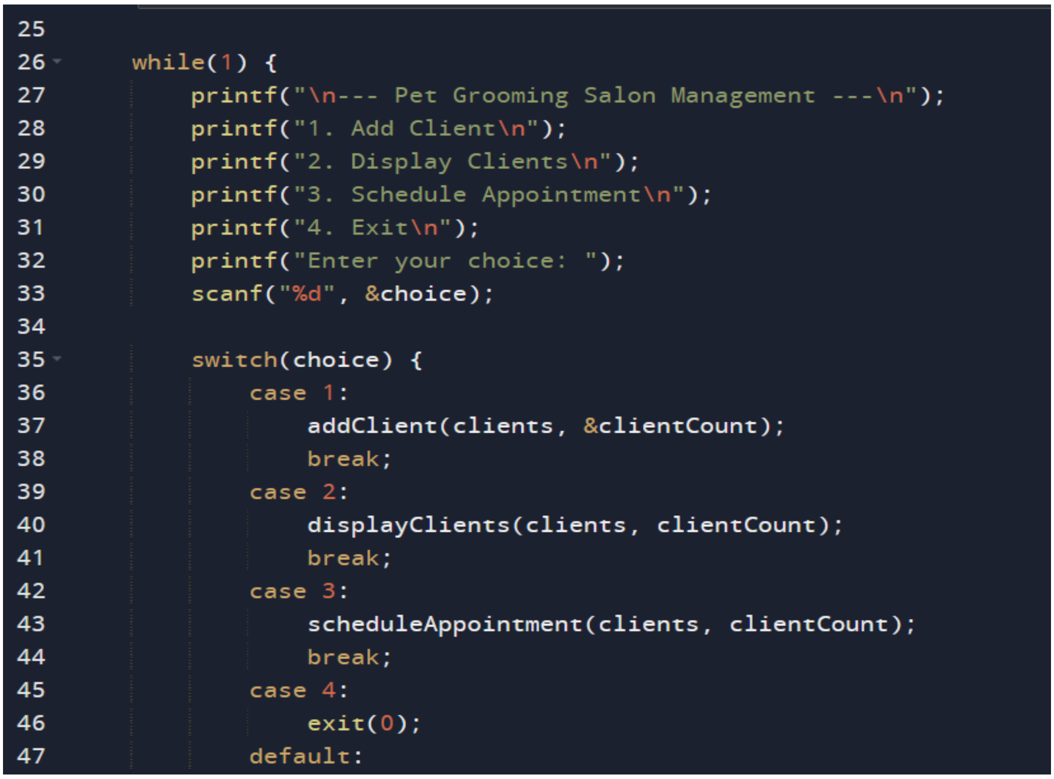
- Online tutorials and documentation on C programming.

Appendices

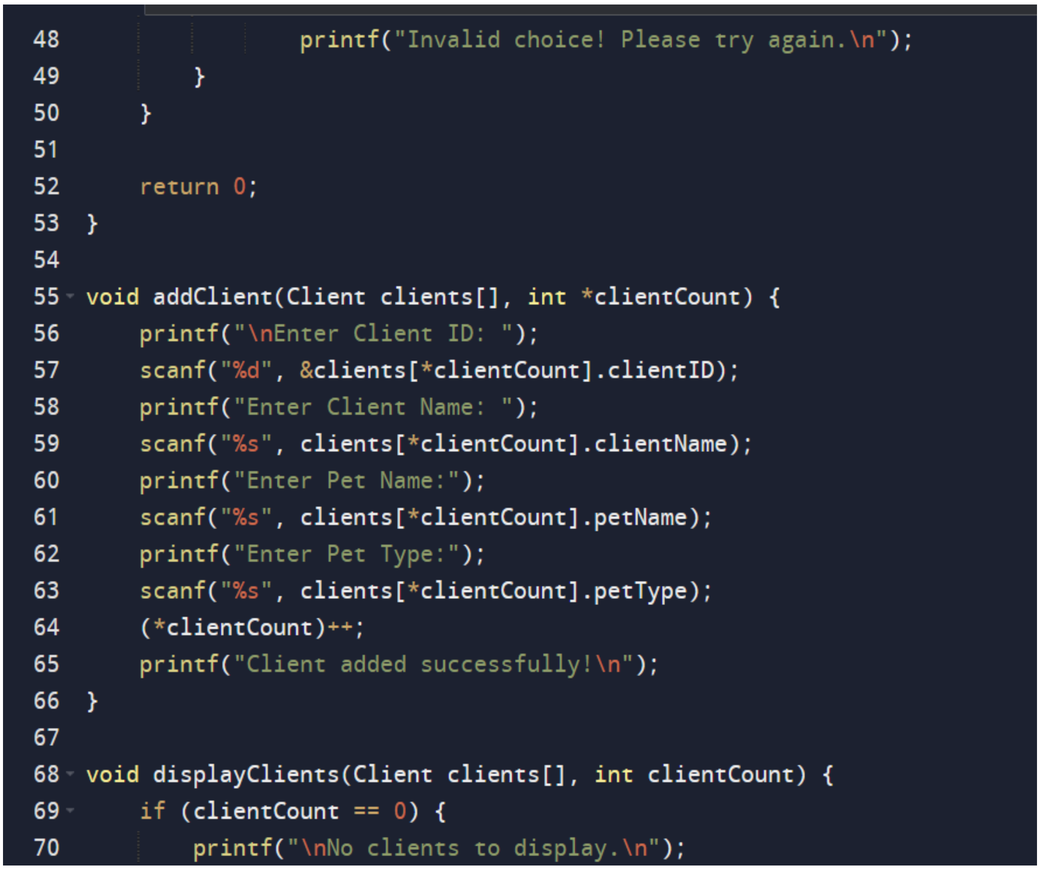
Code Listing



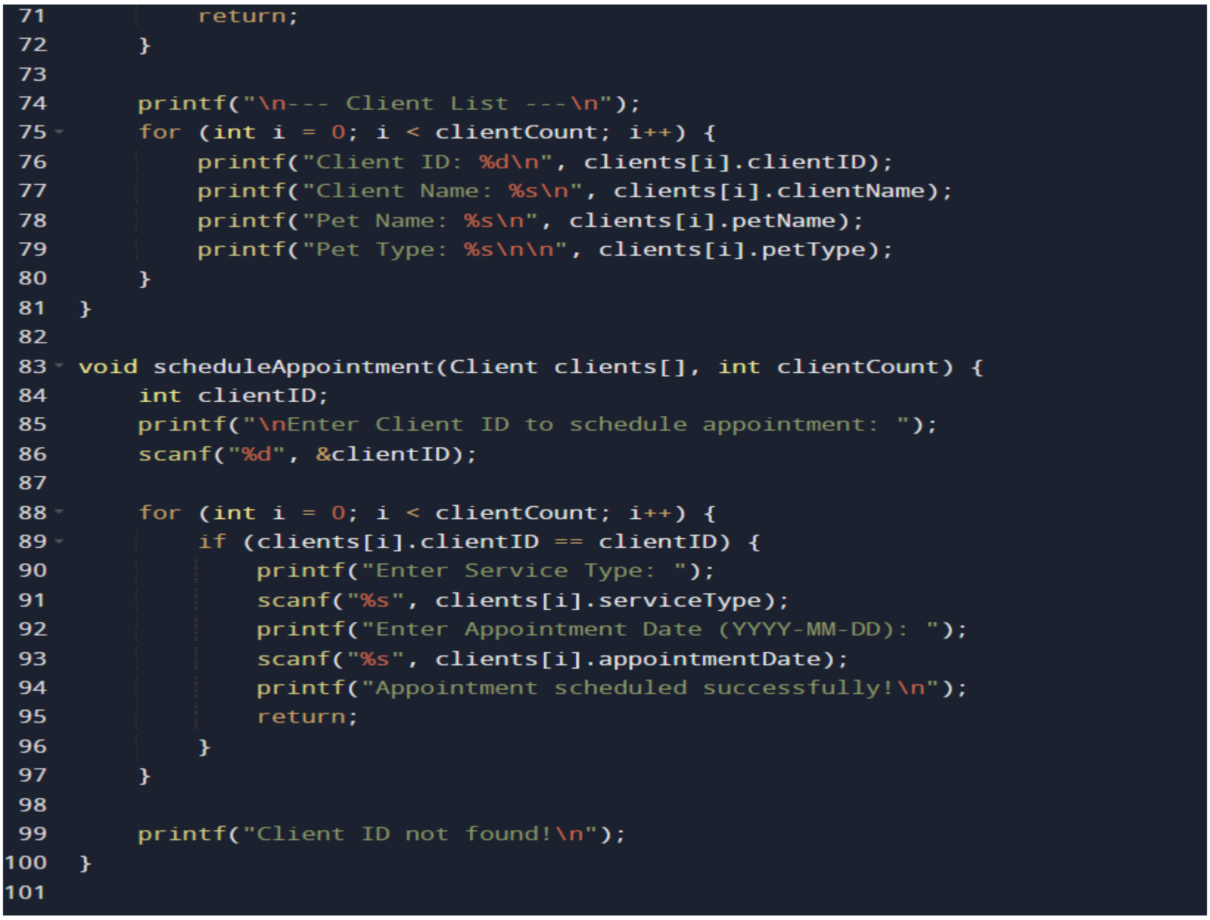
```
// Online C compiler to run C program online
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Structure to store client details
typedef struct {
    int clientID;
    char clientName[50];
    char petName[50];
    char petType[50];
    char serviceType[50];
    char appointmentDate[20];
} Client;

// Function prototypes
void addClient(Client clients[], int *clientCount);
void displayClients(Client clients[], int clientCount);
void scheduleAppointment(Client clients[], int clientCount);

int main() {
    int choice;
    Client clients[100];
    int clientCount = 0;
```



```c

    while(1) {
        printf("\n--- Pet Grooming Salon Management ---\n");
        printf("1. Add Client\n");
        printf("2. Display Clients\n");
        printf("3. Schedule Appointment\n");
        printf("4. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch(choice) {
            case 1:
                addClient(clients, &clientCount);
                break;
            case 2:
                displayClients(clients, clientCount);
                break;
            case 3:
                scheduleAppointment(clients, clientCount);
                break;
            case 4:
                exit(0);
            default:
```



```c
                printf("Invalid choice! Please try again.\n");
        }
    }

    return 0;
}

void addClient(Client clients[], int *clientCount) {
    printf("\nEnter Client ID: ");
    scanf("%d", &clients[*clientCount].clientID);
    printf("Enter Client Name: ");
    scanf("%s", clients[*clientCount].clientName);
    printf("Enter Pet Name:");
    scanf("%s", clients[*clientCount].petName);
    printf("Enter Pet Type:");
    scanf("%s", clients[*clientCount].petType);
    (*clientCount)++;
    printf("Client added successfully!\n");
}

void displayClients(Client clients[], int clientCount) {
    if (clientCount == 0) {
        printf("\nNo clients to display.\n");
```



```
        return;
    }

    printf("\n--- Client List ---\n");
    for (int i = 0; i < clientCount; i++) {
        printf("Client ID: %d\n", clients[i].clientID);
        printf("Client Name: %s\n", clients[i].clientName);
        printf("Pet Name: %s\n", clients[i].petName);
        printf("Pet Type: %s\n\n", clients[i].petType);
    }
}

void scheduleAppointment(Client clients[], int clientCount) {
    int clientID;
    printf("\nEnter Client ID to schedule appointment: ");
    scanf("%d", &clientID);

    for (int i = 0; i < clientCount; i++) {
        if (clients[i].clientID == clientID) {
            printf("Enter Service Type: ");
            scanf("%s", clients[i].serviceType);
            printf("Enter Appointment Date (YYYY-MM-DD): ");
            scanf("%s", clients[i].appointmentDate);
            printf("Appointment scheduled successfully!\n");
            return;
        }
    }

    printf("Client ID not found!\n");
}
```