CSE101: Computer Programming

(Mini Project)

"Ice-cream Parlour Management System"

Registration Number	Roll Number	Name
12221622	Adluri Samhitha sree	RKOCABR29
12221807	Vanamala Durga Prasad	RKOCBRA31
12221657	Aditya Singh Lodhi	RKOCBRA30

12221832	Divyanshu Singh	RKOCBRA33

Submitted to: Parshotam Sir

Index

INTRODUCTION

This project on an Ice cream parlour management system, has been created using C programming language. It aims on making several processes related to the normal management of an ice cream parlour easier and more effective for the owners and the customers.

The ice cream parlour management system is a user friendly and efficient solution to manage ice cream parlours' operations. It helps in minimizing manual error, improving inventory management, and enhancing customer satisfaction by providing accurate and up to date information on ice-cream flavours and prices. The projects implementation using C programming language ensures a good and stable system that can easily be maintained and updated.

Users of the system can maintain and update their inventory, including adding new items and changing the prices of existing ones. The automatic billing system also makes it simple and quick for clients to pay for their products. Additionally, the system offers a simple way to manage customer orders and preferences, including their preferred toppings and flavours.

The project has a straightforward interface and is meant to be user-friendly and simple to use. It may be quickly modified to meet unique business requirements and is appropriate for ice cream parlours of any size, from small shops to major franchises.

In conclusion, our ice cream parlour management system offers a complete set of tools to manage and optimise ice cream parlour operations. It is a dependable and effective solution. It provides excellent value for the money and can greatly enhance business performance.

MODULES EXPLANATION

The Ice cream parlour management system project consists of the following modules:

- **Display Ice cream list**: The user can examine a list of all the ice cream flavours that are offered, along with their respective costs, by using the Display Ice

Cream List module. The system's database can be expanded with additional ice cream flavours and their associated costs by using the Add additional Ice Cream Data module.

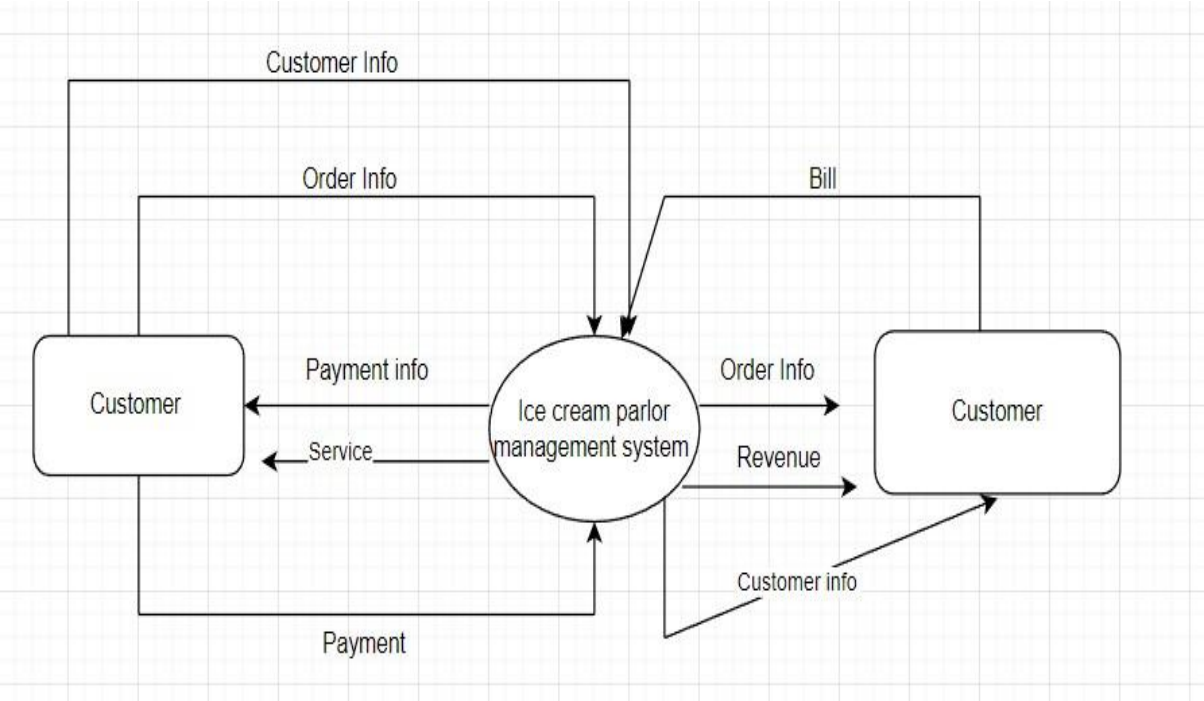
- **Update Record**: The user can change the name and pricing of an existing ice cream flavour by using the Update the Record of the Ice Cream module. The system's database can be searched for a specific ice cream flavour by entering its name in the Search Any Ice Cream module.

- **Add new ice- cream data**: This module allows users to create a new ice cream flavour and to add the data regarding that flavour into the database. The recipe can be added, the temperature can also be added, all the necessary data regarding that particular ice cream flavour can be added
- **Search any Ice-cream**: This module allows users to search the particular flavour of ice cream they desire, this will make the process of locating a particular flavour much more easier.

- **Delete any ice-cream Record**: This module allows users to delete any existing record related to any ice cream flavour, user can also delete any undesired data from their accounts and make it according to their needs.

Each module in this management system project has its own significance and it is the sole purpose of this project to integrate all these modules and make the process of ordering ice-cream much more simpler yet effective.

DATA FLOW DIAGRAM



CODE

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

```
#include <stdlib.h>
#include <string.h>
struct Icecream{
char name[50];
int price;
};
struct DisplayIcecream{
char name[50];
int price;
};
void AddNewIcecream() {
struct Icecream icecream;
printf("Enter Icecream name: vanilla ");
scanf("%c",icecream.name);
printf("Enter the Icecream price: ");
scanf("%d", icecream.price);
FILE *file = fopen("data.txt", "a");
fprintf(file, "Icecream|%c|%d|\n", icecream.name, icecream.price
);
fclose(file); }
void addIcecream() {
struct Icecream icecream;
printf("\nEnter icecream name: vanilla");
scanf("%c", icecream.name);
```

```
printf("\nEnter icecream price: ");
scanf("%d", &icecream.price);
FILE *file = fopen("data.txt", "a");
fprintf(file, "Icecream|%c|%d\n", icecream.name, icecream.price
);
fclose(file);
}
void displayIcecream() {
FILE *file = fopen("data.txt", "r");
char buffer[100];
printf("Name\tPrice\n");
while (fgets(buffer, 100, file) != NULL) {
char *token = strtok(buffer, "|");
if (strcmp(token, "Icecream") == 0) {
printf("%s\t", strtok(NULL, "|"));
printf("%s\t", strtok(NULL, "|"));
}
}
fclose(file);
}
void DisplayIcecream() {
FILE *file = fopen("data.txt", "r");
char buffer[100];
printf("Name\tPrice\n:Vanilla\t300");
```

```
while (fgets(buffer, 100, file) != NULL) {
char *token = strtok(buffer, "|");
if (strcmp(token, "Icecream") == 0) {
printf("%s\t", strtok(NULL, "|"));
printf("%s\n", strtok(NULL, "|"));
}
}
fclose(file);
}
void deleteIcecream() {
char name[50];
printf("Enter icecream name to delete: ");
scanf("%c", name);
FILE *file = fopen("data.txt", "r");
FILE *temp = fopen("temp.txt", "w");
char buffer[100];
int found = 0;
while (fgets(buffer, 100, file) != NULL) {
char *token = strtok(buffer, "|");
if (strcmp(token, "Icecream") == 0 && strcmp(strtok(NULL,
"|"), name) == 0) {
found = 1;
} else {
fprintf(temp, "%c", buffer);
```

```
}
}
fclose(file);
fclose(temp);
remove("data.txt");
rename("temp.txt", "data.txt");
if (found) {
printf("Icecream deleted successfully!\n");
} else {
printf("Icecream not found...\n");
}
}
void DeleteIcecream() {
char name[50];
printf("Enter icecream name to delete: ");
scanf("%c", name);
FILE *file = fopen("data.txt", "r");
FILE *temp = fopen("temp.txt", "w");
char buffer[100];
int found = 0;
while (fgets(buffer, 100, file) != NULL) {
char *token = strtok(buffer, "|");
if (strcmp(token, "Icecream") == 0 && strcmp(strtok(NULL, "|"),
name)
== 0) {
```

```
found = 1;
} else {
fprintf(temp, "%c", buffer);
}
}
fclose(file);
fclose(temp);
remove("data.txt");
rename("temp.txt", "data.txt");
if (found) {
printf("Icecream deleted successfully!\n");
} else {
printf("Icecream not found...\n");
}
}
void updateIcecream() {
char name[50];
printf("Enter candidate name to update: ");
scanf("%c", name);
FILE *file = fopen("data.txt", "r");
FILE *temp = fopen("temp.txt", "w");
char buffer[100];
int found = 0;
while (fgets(buffer, 100, file) != NULL) {
```

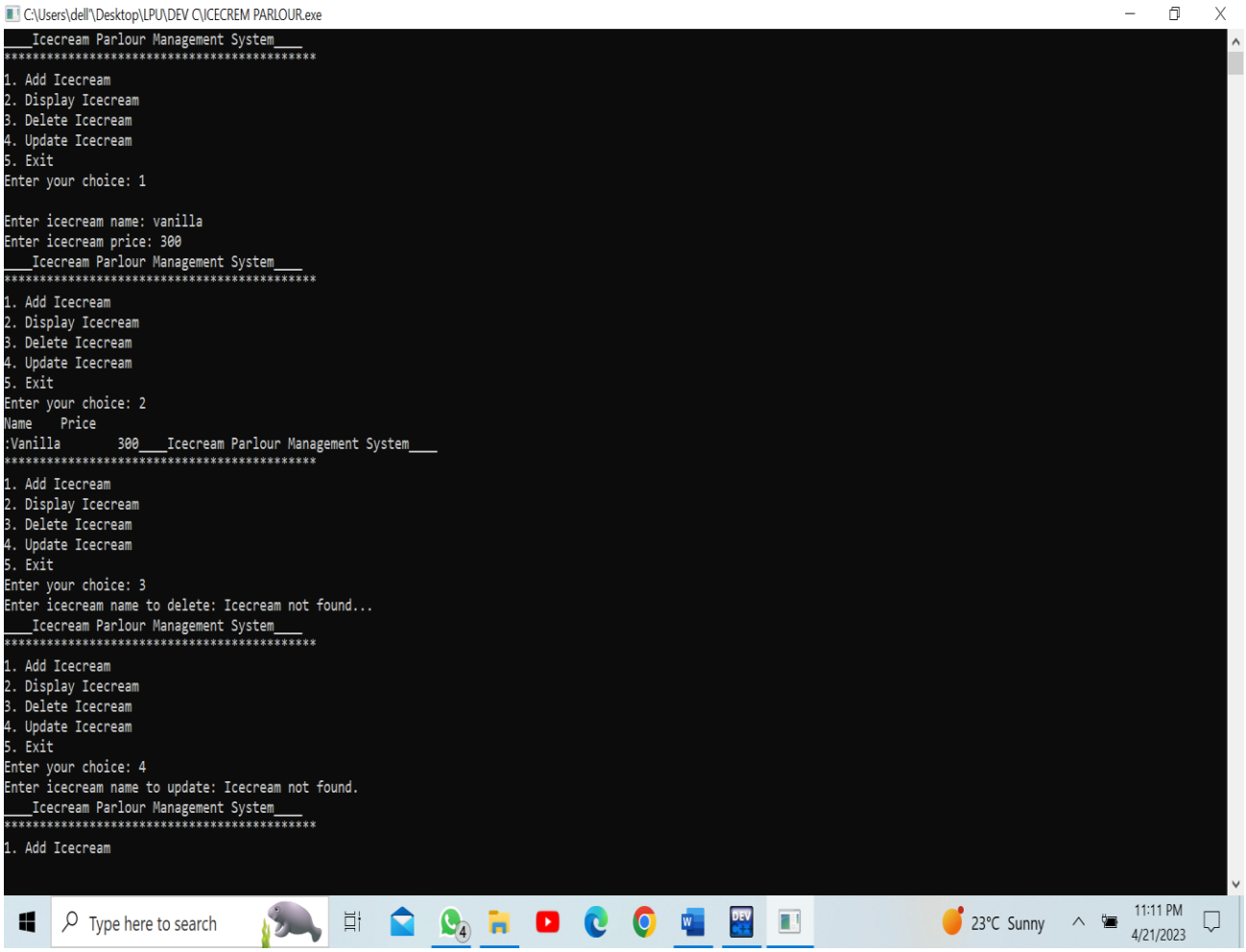
```
char *token = strtok(buffer, "|");
if (strcmp(token, "Icecream") == 0 && strcmp(strtok(NULL,
"|"), name) == 0) {
found = 1;
struct Icecream icecream;
printf("Enter updated icecream name: ");
scanf("%c", icecream.name);
printf("Enter updated icecream price: ");
scanf("%d", icecream.price);
fprintf(temp, "Icecream|%c|%d\n", icecream.name, icecream.price
);
} else {
fprintf(temp, "%s", buffer);
}
}
fclose(file);
fclose(temp);
remove("data.txt");
rename("temp.txt", "data.txt");
if (found) {
printf("Icecream updated successfully!\n");
} else {
printf("Icecream not found...\n");
}
```

```
}
void UpdateIcecream() {
char name[50];
printf("Enter icecream name to update: ");
scanf("%c", name);
FILE *file = fopen("data.txt", "r");
FILE *temp = fopen("temp.txt", "w");
char buffer[100];
int found = 0;
while (fgets(buffer, 100, file) != NULL) {
char *token = strtok(buffer, "|");
if (strcmp(token, "Icecream") == 0 && strcmp(strtok(NULL, "|"),
name)
== 0) {
found = 1;
struct Icecream icecream;
printf("Enter updated icecream name: ");
scanf("%c", icecream.name);
printf("Enter updated icecream price: ");
scanf("%d", &icecream.price);
fprintf(temp, "Icecream|%c|%d\n", icecream.name, icecream.price
);
} else {
fprintf(temp, "%c", buffer);
}
```

```
}
fclose(file);
fclose(temp);
remove("data.txt");
rename("temp.txt", "data.txt");
if (found) {
printf("Icecream updated successfully!\n");
} else {
printf("Icecream not found.\n");
}
}
int main() {
int choice;
do {
printf("____Icecream Parlour Management System____\n");
printf("*****************************************\n");
printf("1. Add Icecream\n");
printf("2. Display Icecream\n");
printf("3. Delete Icecream\n");
printf("4. Update Icecream\n");
printf("5. Exit\n");
printf("Enter your choice: ");
scanf("%d", &choice);
switch (choice)
```

```
{
        case 1:
                addIcecream();
                break;
                case 2:
                        DisplayIcecream();
                        break;
                        case 3:
                                DeleteIcecream();
                                break;
                                case 4:
                                        UpdateIcecream();
                                        break;
                                        case 5:
                                                printf("Goodbye\n");
                                                break;
                                                default:
                                                        printf("Invalid choice\n");
   }
} while(choice!=5);
return 0;
}
```

OUTPUT SCREENSHOT



THANK YOU