

**College of Computer Sciences and Information Technology**

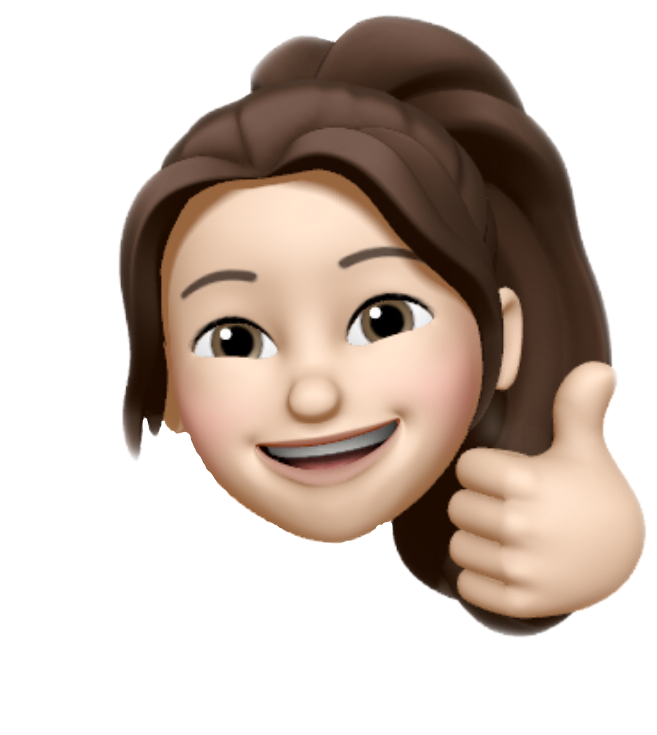
**King Faisal University**

**Fundamentals of Programming**

**Final Project**

***Project title:***

**Beauty Salon Reservation System**



***Member:***

**Hissah Almuhaysh - 222433855**

***Submitted to:***

Dr. Sharmila Banu Sheik Imam

***Date:***

**February - 2023**

**Fundamentals Of Programming**

**2022-2023 (semester 2)**

**section: 68**

TABLE OF CONTENTS

[**I. PROJECT DESCRIPTION**](#_heading=h.ovify6rqau0g) **2**

[**II. PROGRAM FEATURES**](#_heading=h.cpjit84vkl) **3**

[**II. USER MANUAL**](#_heading=h.3j6v1yde5z6r) **4**

[**IV. PROGRAM CODE**](#_heading=h.uaxj14hc7h8n) **9**

# 

# 

# 

# 

# 

# I. PROJECT DESCRIPTION

This project is a salon management system designed to help hairdressers to have all the information and details of any customers in the salon, in an easier way that allows more time and effective ways of working from the perspective of care and administrative elements. Move from paper-based documents or spreadsheets with information and details intended for hairdressers or stylists to a more accessible, cheaper, and more organized way with this upgraded salon management system. This system allows all data to be stored and accessed in one place that is easily accessible at any time, and collecting all these elements is what we aim to achieve in this project.

# II. PROGRAM FEATURES

* **Add a new customer**

This feature allows the user to add the customer details such as (name, phone, address, preferable hairstyle).

* **Add a new hairstylist**

This feature allows the user to add the hairstylist's details such as (name, phone, specialty, nationality).

* **Add a new reservation**

This feature allows you to assign any customer reservation and choose a hairstylist.

* **Cancel reservation**

This feature allows you to remove any customer reservation through their customer details.

* **Update customer details**

This feature allows the user to update any customer details through the name of the customer.

* **Update hairstylist details**

This feature allows the user to update any hairstylist details through the hairstylist's name.

* **Delete customer records**

This feature allows the user to delete any customer records through the name of the customer.

* **Delete hairstylist records**

This feature allows the user to delete any hairstylist records through the hairstylist's name.

* **Show customer lists**

This feature allows the user to display all the customers listed in the Salon and their details.

* **Show hairstylist lists**

This feature allows the user to display all the hairstylists listed in the Salon and their details.

* **Search for specific customer details**

This feature allows the user to search for details about the customer and includes the hairstylist's details.

* **Search for available hairstylists**

This feature allows the user to search for details about the hairstylist and includes customer details.

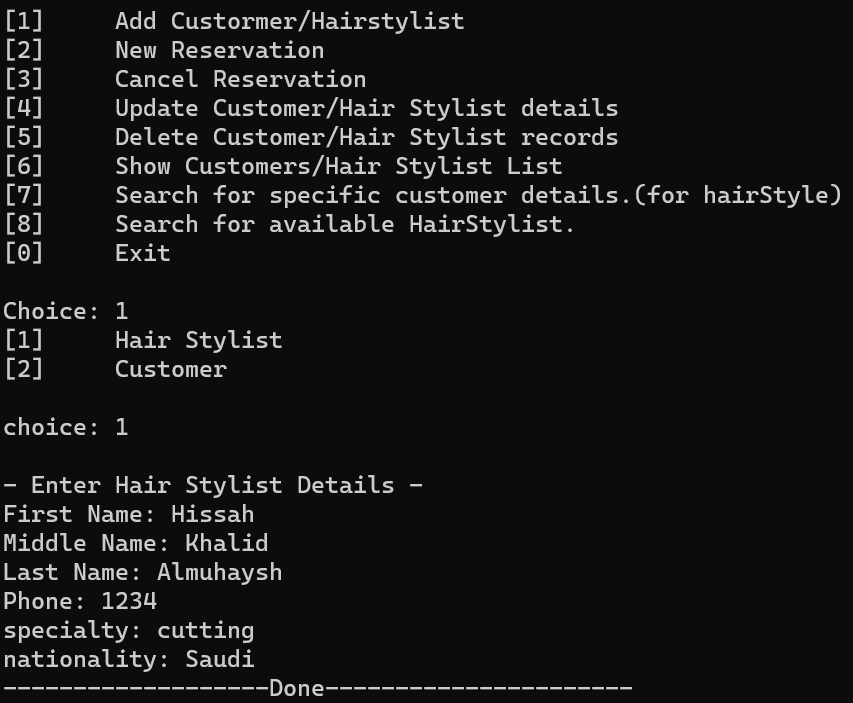
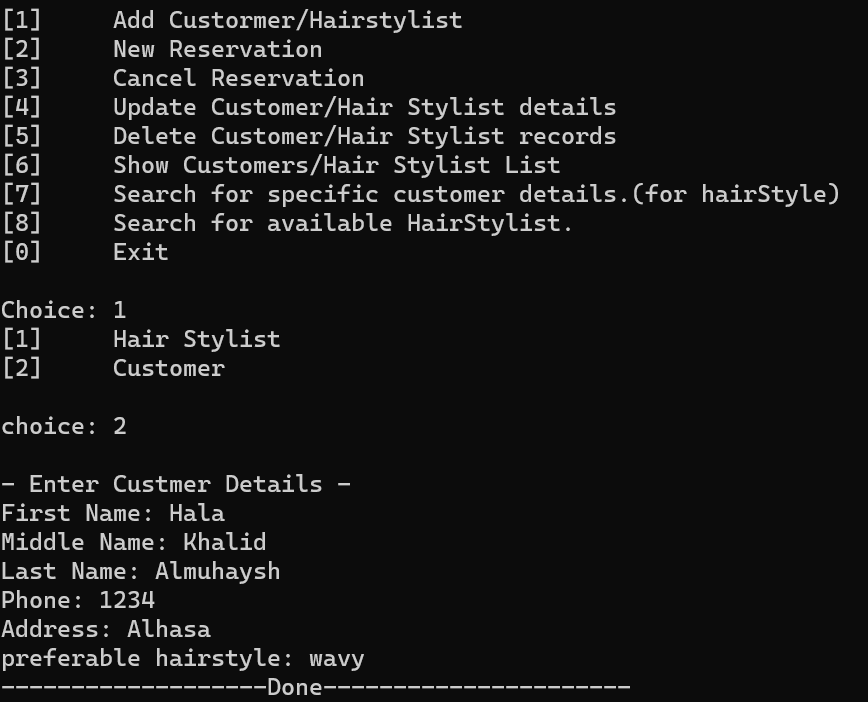
* **Exit**

This feature allows the salon management system to exit successfully.

# 

# II. USER MANUAL

#Add Customer/Hairstylist

The salon system software is very easy to use, we start by reading the selections presented in the software output below and then we start using our software completely, and to do this you must complete two steps:

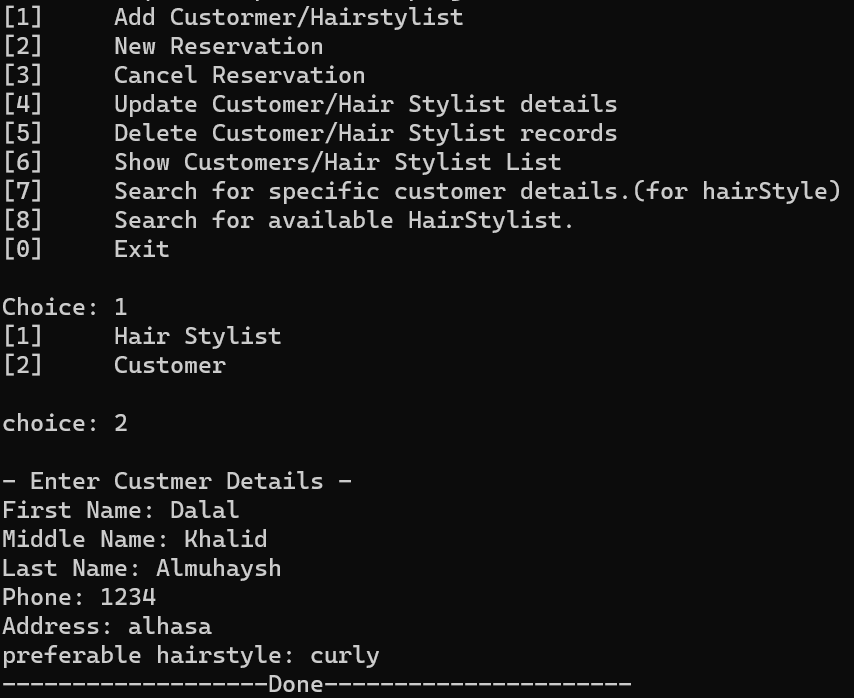
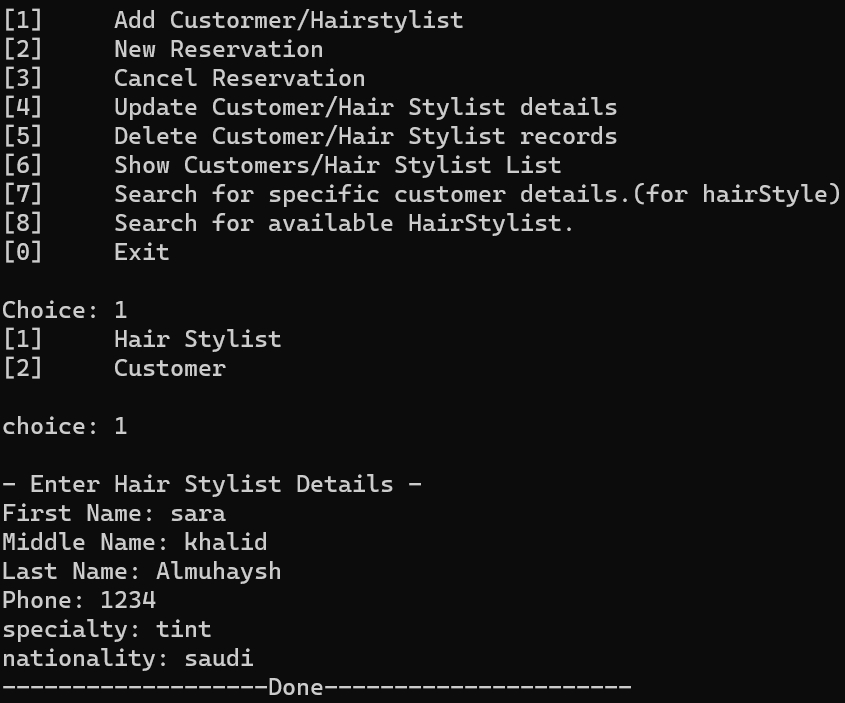
**1**

**2**

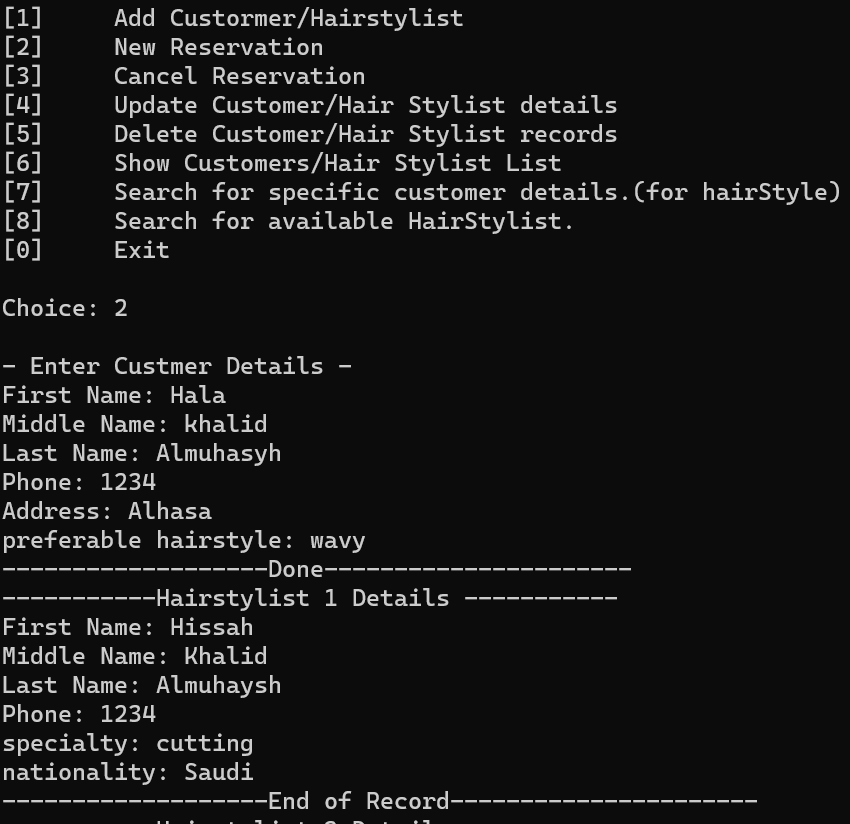
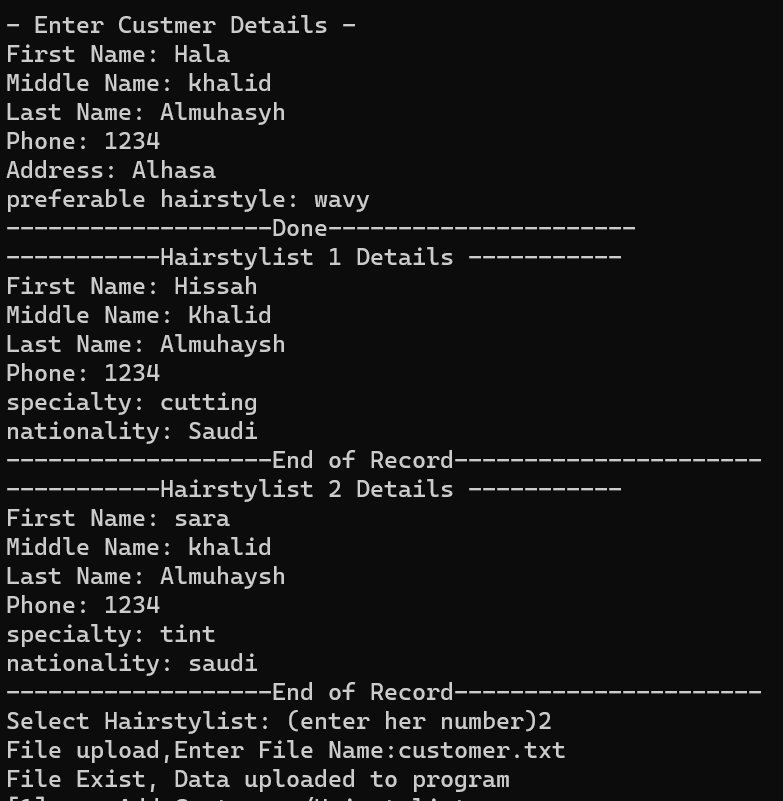
**3**

**4**

In the first box, this step is if you want to add a customer or a hairstylist. In the second box, you have to write the number 1 to open the third box for you. It contains a mini-menu to choose to add customer or hairstylist information. The fourth box to add the hairstylist's information (first name, middle name, last name, phone, specialty, nationality) You can add more than one hairstylist or o add the customer's information (first name, middle name, last name, phone, address, preferable hairstyle) You can add more than one customer.

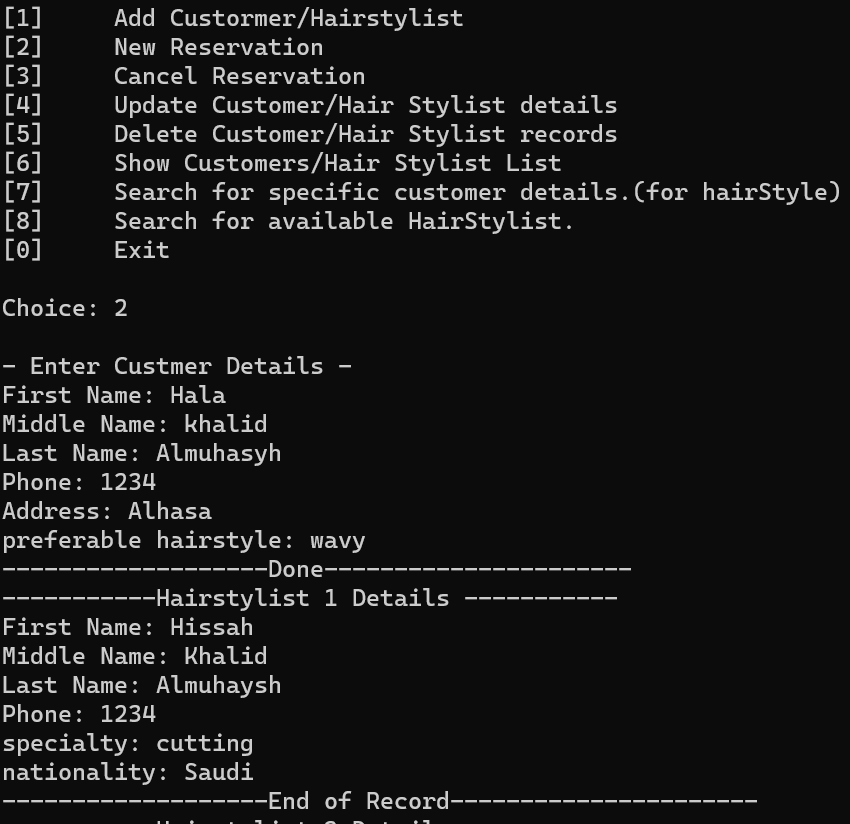
**-Below are examples repeating the same procedure above**

#Add New Reservation

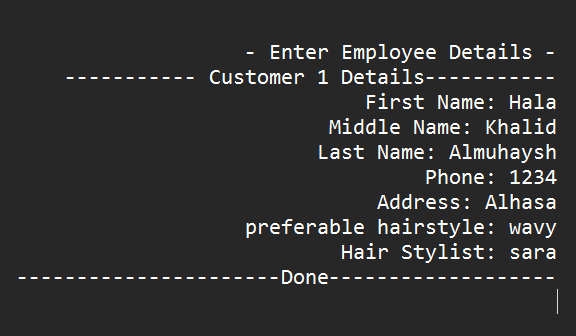
In the first box, add a reservation using the customer's information. In the second box, you need to write the number 2 so that the option opens. In the third box, you write the new customer's information and choose the hairstylist using his number in the list. In the fourth box, you write the name of the file that you want to store the reservation.

**3**

**1**

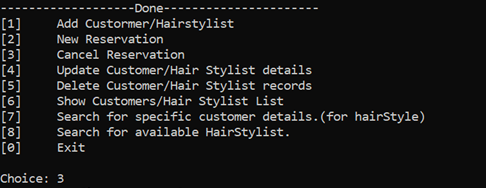
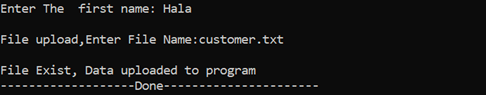


**2**



**4**

cancel the reservation#

In the first box, cancel a reservation using the customer's name, in the second box, you need to write the number 3 so that the option opens, in the third box, you should write the customer's name, int the fourth box, you write the same file name in which you store the reservation.

**1**

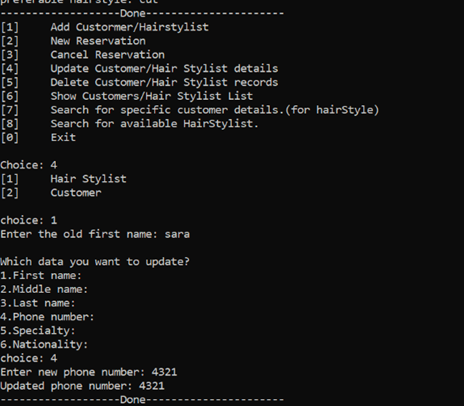
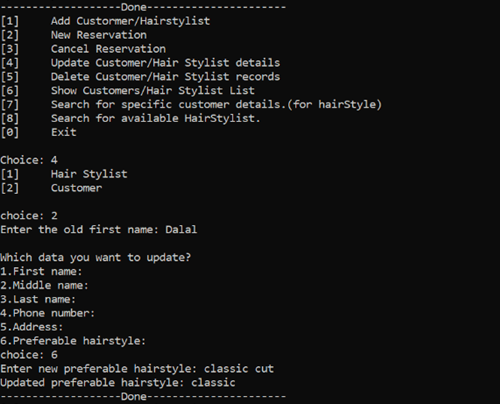
**2**

**3**

**4**

Update Customer/Hair Stylist records#

In the fourth box, update the customer or hairstylist's information, int the second box, you should write 4, so that a mini-menu opens for you in the third box, choose whether you want to update a customer or hairstylist, and fourth box, you write who you want to update by his first name, in the fifth box, choose the data you want to update, in the sixth box, enter the new data.



**1**

**6**

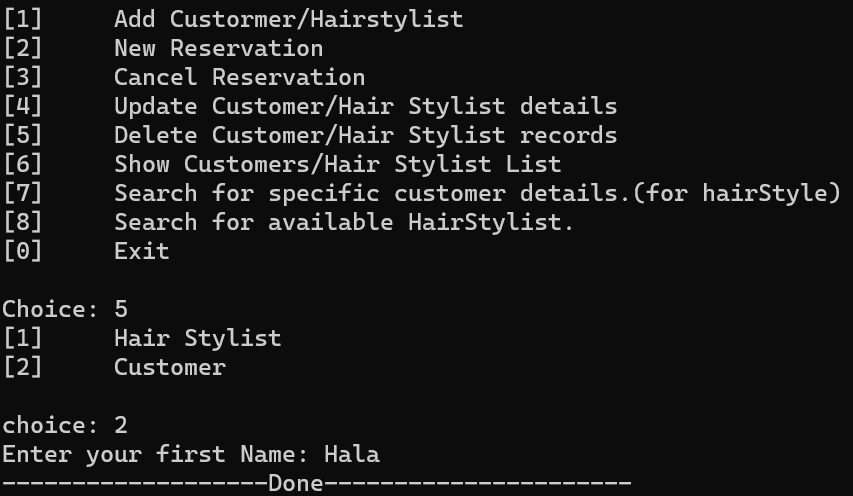
**5**

**4**

**3**

**2**

#Delete Customer/Hair Stylist records

In the first box, delete the customer or hairstylist's information, in the second box, you should write 5, so that a mini-menu opens for you in the third box, choose whether you want to delete a customer or hairstylist, and in the fourth box, you write who you want to delete by his first name.

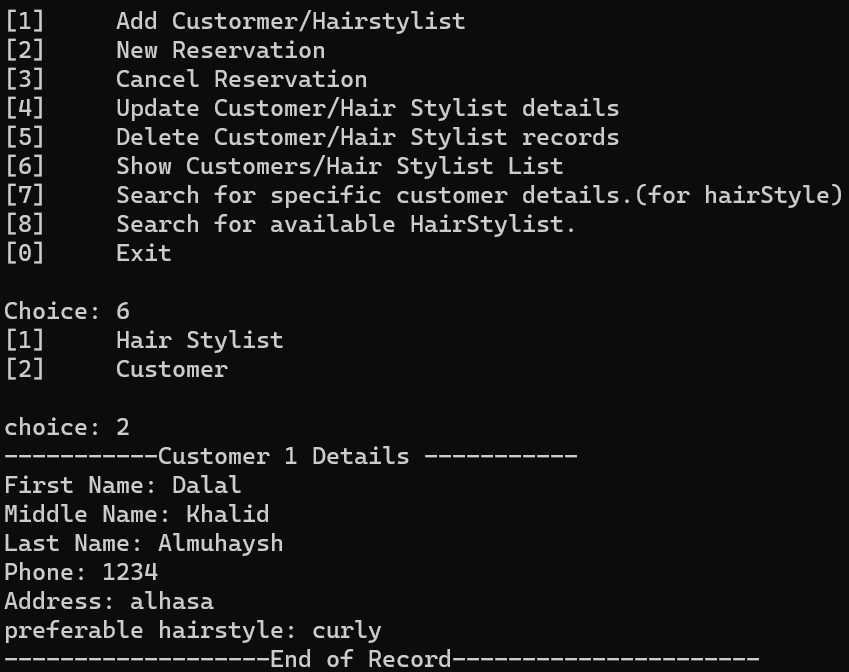
**1**

**2**

**3**

**4**

#Show Customers/Hair Stylist List

In the first box, show a list of buyers and hairdressers, and in the second box, you must type 6, so that a mini-list opens for you. In the third box, choose whether you want to see clients or hairdressers.

**1**

**2**

**3**

#search for a specific customer.

In the first box, search for a specific customer, and in the second box, you must type 7, In the third box, enter the name of the customer you want to search for, then it will display customer details.



**1**

**2**

**3**

#search for available hairstyles

In the first box, search for available hairstyles, and in the second box, you must type 8, In the third box, enter the name of the hairstyle you want to search, then it will display hairstyle details.

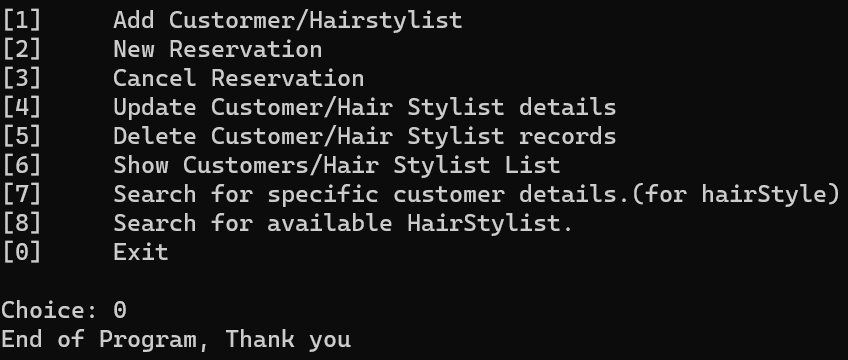
**2**

**1**

**3**

#Exit

In the first box, the command to exit the program, the second box, you have to write 0 to exit the salon system successfully.



**1**

**2**

# IV. PROGRAM CODE

[click here to open the source](https://drive.google.com/file/d/1SkgKlL1VoVutF-JSxS1cVHjP4DdA7czB/view?usp=sharing)

#include <iostream>

#include <fstream> //Header files we used.

#include <string>

# define Size 1000

using namespace std;

#define Customer\_COUNT 10

#define Hairstylist\_COUNT 10

struct NameType

{

string first;

string middle;

string last;

};

struct AddressType

{

string address1;

};

//structure for CustomerType record.

struct CustomerType

{

NameType name;

AddressType address;

int number;

string preferable;

}customerList[Customer\_COUNT];

struct reservation{

NameType name;

string address;

int number;

string preferable;

string hairstylist;

}reservationList[50];

//add function for Customer.

void addCustomer(int count)

{

cout << "\n- Enter Customer Details -\n";

cout << "First Name: ";

cin >> customerList[count].name.first;

cout << "Middle Name: ";

cin >> customerList[count].name.middle;

cout << "Last Name: ";

cin >> customerList[count].name.last;

cout << "Phone: ";

cin >> customerList[count].number;

cout << "Address: ";

cin >> customerList[count].address.address1;

cout << "preferable hairstyle: ";

cin >> customerList[count].preferable;

cout<< "-------------------Done----------------------\n";

}

struct nameHairstylistType

{

string first;

string middle;

string last;

};

struct Hairstylistspecialty

{

string specialty1;

};

//structure for HairstylistType record.

struct HairstylistType

{

nameHairstylistType name;

Hairstylistspecialty specialty;

int number;

string nationality;

}HairstylistList[Hairstylist\_COUNT];

//add function for Hairstylist.

void addHairstylist(int count)

{

cout << "\n- Enter Hairstylist Details -\n";

cout << "First Name: ";

cin >> HairstylistList[count].name.first;

cout << "Middle Name: ";

cin >> HairstylistList[count].name.middle;

cout << "Last Name: ";

cin >> HairstylistList[count].name.last;

cout << "Phone: ";

cin >> HairstylistList[count].number;

cout << "specialty: ";

cin >> HairstylistList[count].specialty.specialty1;

cout << "nationality: ";

cin >> HairstylistList[count].nationality;

cout<< "-------------------Done----------------------\n";

}

//disply function for Hairstyl.

void displayHairstylist(int ctr)

{

int found=0;

for(int count = 0; count < ctr; count++){

cout << "-----------Hairstylist " << count + 1 <<" Details -----------\n";

cout << "First Name: " << HairstylistList[count].name.first << "\n";

cout << "Middle Name: " << HairstylistList[count].name.middle << "\n";

cout << "Last Name: " << HairstylistList[count].name.last << "\n";

cout << "Phone: " << HairstylistList[count].number << "\n";

cout << "specialty: " << HairstylistList[count].specialty.specialty1<< "\n";

cout << "nationality: " << HairstylistList[count].nationality << "\n";

cout << "-------------------End of Record----------------------\n";

found++;

}

if(found==0){

cout<<"-------------------Hairstylist not found!-------------------\n";

}

}

//display function for reservation.

void displayreservation(int ctr)

{

for(int count = 0; count < ctr; count++){

cout << "-----------Hairstylist " << count + 1 <<" Details -----------\n";

cout << "First Name: " << reservationList[count].name.first << "\n";

cout << "Middle Name: " << reservationList[count].name.middle << "\n";

cout << "Last Name: " << reservationList[count].name.last << "\n";

cout << "Address: " << reservationList[count].address << "\n";

cout << "Phone: " << reservationList[count].number << "\n";

cout << "preferable: " << reservationList[count].preferable<< "\n";

cout << "hairstylist: " << reservationList[count].hairstylist << "\n";

cout << "-------------------End of Record----------------------\n";

}

}

//display function for Customer.

void displaycustomer(int ctr)

{

int found=0;

for(int count = 0; count < ctr; count++){

cout << "-----------Customer " << count + 1 <<" Details -----------\n";

cout << "First Name: " << customerList[count].name.first << "\n";

cout << "Middle Name: " << customerList[count].name.middle << "\n";

cout << "Last Name: " << customerList[count].name.last << "\n";

cout << "Phone: " << customerList[count].number << "\n";

cout << "Address: " << customerList[count].address.address1<< "\n";

cout << "preferable hairstyle: " << customerList[count].preferable << "\n";

cout << "-------------------End of Record----------------------\n";

found++;

}

if(found==0){

cout<<"-------------------not found-------------------\n";

}

}

//delete function for Hairstyl record by name.

void deleteHairstylist(string name,int counter){

int found;

for(int ctr = 0; ctr < counter; ctr++)

{

if(name == HairstylistList[ctr].name.first)

{

for(int i = ctr; i < counter; i++){

HairstylistList[i].name.first= HairstylistList[i+1].name.first ;

HairstylistList[i].name.middle= HairstylistList[i+1].name.middle ;

HairstylistList[i].name.last= HairstylistList[i+1].name.last ;

HairstylistList[i].number= HairstylistList[i+1].number ;

HairstylistList[i].specialty.specialty1= HairstylistList[i+1].specialty.specialty1;

HairstylistList[i].nationality= HairstylistList[i+1].nationality;

}

cout<< "-------------------Done----------------------\n";

found=1;

break;

}

}

if(found==0){

cout<< "-------------------not found!------------------- "<<endl;

}

}

//delete function for Customer record by name.

void deleteCustomer(string name ,int counter){

int found;

for(int ctr = 0; ctr < counter; ctr++)

{

if(name == customerList[ctr].name.first)

{

for(int i = ctr; i < counter; i++){

customerList[i].name.first= customerList[i+1].name.first ;

customerList[i].name.middle= customerList[i+1].name.middle ;

customerList[i].name.last= customerList[i+1].name.last ;

customerList[i].number= customerList[i+1].number ;

customerList[i].address.address1= customerList[i+1].address.address1;

customerList[i].preferable= customerList[i+1].preferable;

}

cout<< "-------------------Done----------------------\n";

break;

}

}

if(found==0){

cout<< "-------------------not found!------------------- "<<endl;

}

}

//function for file.

void FileLoad(int counter, int H)

{

int ListSize=0;

ifstream input;

ofstream output;

char FileName[50];

H--;

cout<<"File upload,Enter File Name:";

cin>>FileName;

output.open(FileName);

if (!output.is\_open())

cout << " problem with opening the file ";

else

{

while (!output.eof())

{

ofstream output(FileName);

for(int ctr = 0; ctr < counter; ctr++)

{

reservationList[ctr].hairstylist=HairstylistList[H].name.first;

output << "\n- Enter Customer Details -\n";

output << "-----------Customer " << ctr + 1 <<" Details -----------\n";

output << "First Name: " << reservationList[ctr].name.first << "\n";

output << "Middle Name: " << reservationList[ctr].name.middle << "\n";

output << "Last Name: " << reservationList[ctr].name.last << "\n";

output << "Phone: " << reservationList[ctr].number << "\n";

output << "Address: " << reservationList[ctr].address<< "\n";

output << "preferable hairstyle: " << reservationList[ctr].preferable << "\n";

output << "Hair Stylist: "<<reservationList[ctr].hairstylist<<" \n";

output<< "-------------------Done----------------------\n";

ListSize++;

}

output.close();

break;

}

cout<<"\nFile Exist, Data uploaded to program\n";

cout<< "-------------------Done----------------------\n";

input.close();

}

}

//search function for Customer by name.

void searchCustomer(string name, int counter ){

int found;

for(int ctr = 0; ctr < counter; ctr++){

if(name == customerList[ctr].name.first){

displaycustomer(counter);

found =1;

break;

}

}

if(found==0){

cout<< "Customer not found!"<<endl;

}

}

//search function for Hairstyl by name.

void searchHairstylist(string name, int counter ){

int found;

for(int ctr = 0; ctr < counter; ctr++){

if(name == HairstylistList[ctr].name.first){

displayHairstylist(counter);

found =1;

break;

}

}

if(found==0){

cout<< "Hairstylist not found!"<<endl;

}

}

//update function for Hairstyl by name.

void updatehairstylist(int count){

int num, found;

string name;

for(int i = 0; i < count ; i++){

if(name==HairstylistList[i].name.first){

found=1;

cout << "\nWhich data you want to update?\n";

cout<<"1.First name: "<<endl;

cout<<"2.Middle name: "<<endl;

cout<<"3.Last name: "<<endl;

cout<<"4.Phone number: "<<endl;

cout<<"5.Specialty: "<<endl;

cout<<"6.Nationality: "<<endl;

cout<<"choice: ";

cin>>num;

switch(num){

case 1:

cout<<"Enter new first name: ";

cin>>HairstylistList[i].name.first;

cout<<"Updated first name: "<<HairstylistList[i].name.first<<endl;

break;

case 2:

cout<<"Enter new middle name: ";

cin>>HairstylistList[i].name.middle;

cout<<"Updated middle name: "<<HairstylistList[i].name.middle<<endl;

break;

case 3:

cout<<"Enter new last name: ";

cin>>HairstylistList[i].name.last;

cout<<"Updated last name: "<<HairstylistList[i].name.last<<endl;

break;

case 4:

cout<<"Enter new phone number: ";

cin>>HairstylistList[i].number;

cout<<"Updated phone number: "<<HairstylistList[i].number<<endl;

break;

case 5:

cout<<"Enter new speciality: ";

cin>>HairstylistList[i].specialty.specialty1;

break;

case 6:

cout<<"Enter new nationality: ";

cin>>HairstylistList[i].nationality;

cout<<"Updated nationality: "<<HairstylistList[i].nationality<<endl;

break;

default:

cout<<"Enter a required number."<<endl;

break;

}

cout<< "-------------------Done----------------------\n";

}

else if(name!=HairstylistList[i].name.first)

cout << "\nHairstylist is not available!\n\n";

}

if(found==0)

cout<<"not found!"<<endl;

}

//update function for Customer by name.

void updatecustomer( int count){

int num, found;

string name;

for(int i = 0; i < count ; i++){

if(name==customerList[i].name.first){

found=1;

cout << "\nWhich data you want to update?\n";

cout<<"1.First name: "<<endl;

cout<<"2.Middle name: "<<endl;

cout<<"3.Last name: "<<endl;

cout<<"4.Phone number: "<<endl;

cout<<"5.Address: "<<endl;

cout<<"6.Preferable hairstyle: "<<endl;

cout<<"choice: ";

cin>>num;

switch(num){

case 1:

cout<<"Enter new first name: ";

cin>>customerList[i].name.first;

cout<<"Updated first name: "<<customerList[i].name.first<<endl;

break;

case 2:

cout<<"Enter new middle name: ";

cin>>customerList[i].name.middle;

cout<<"Updated middle name: "<<customerList[i].name.middle<<endl;

break;

case 3:

cout<<"Enter new last name: ";

cin>>customerList[i].name.last;

cout<<"Updated last name: "<<customerList[i].name.last<<endl;

break;

case 4:

cout<<"Enter new phone number: ";

cin>>customerList[i].number;

cout<<"Updated phone number: "<<customerList[i].number<<endl;

break;

case 5:

cout<<"Enter new address: ";

cin>>customerList[i].address.address1;

cout<<"Updated address: "<<customerList[i].address.address1<<endl;

break;

case 6:

cout<<"Enter new preferable hairstyle: ";

cin>>customerList[i].preferable;

cout<<"Updated preferable hairstyle: "<<customerList[i].preferable<<endl;

break;

default:

cout<<"Enter a required number."<<endl;

break;

}

cout<< "-------------------Done----------------------\n";

}

else if(name!=customerList[i].name.first)

cout << "\nCustomer is not found!\n\n";

}

if(found==0)

cout<<"not found!"<<endl;

}

//add function for reservation.

void addreservation(int count){

cout<<"Enter Customer first name: ";

cin>>reservationList[count].name.first;

cout<<"Enter Customer middle name: ";

cin>>reservationList[count].name.middle;

cout<<"Enter Customer last name: ";

cin>>reservationList[count].name.last;

cout<<"Enter adress: ";

cin>>reservationList[count].address;

cout<<"Enter number: ";

cin>>reservationList[count].number;

cout<<"Enter preferable : ";

cin>>reservationList[count].preferable;

cout<< "-------------------Done----------------------\n";

}

//cancel function for reservation.

// You can Not cancel Reservation berfor add it .

void cancelReservation(int count){

int found=0, j ;

string n1;

cout<<"Enter The first name: ";

cin>>n1;

for(int i=0; i<count; i++)

{

if(reservationList[i].name.first ==n1)

{

for(j=i; j<(count-1); j++)

reservationList[j] = reservationList[j+1];

found++;

i--;

count--;

}

}

if(found==0){

cout<<"Reservation doesn't found !";

cout<<endl;}

if(found!=0){

int ListSize=0;

ifstream input;

ofstream output;

char FileName[50];

cout<<"File upload,Enter File Name:";

cin>>FileName;

output.open(FileName);

if (!output.is\_open())

cout << " problem with opening the file ";

else

{

while (!output.eof())

{

ofstream output(FileName);

for(int ctr = 0; ctr < count; ctr++)

{

output << "\n- Enter Customer Details -\n";

output << "-----------Customer " << ctr + 1 <<" Details -----------\n";

output << "First Name: " << reservationList[ctr].name.first << "\n";

output << "Middle Name: " << reservationList[ctr].name.middle << "\n";

output << "Last Name: " << reservationList[ctr].name.last << "\n";

output << "Phone: " << reservationList[ctr].number << "\n";

output << "Address: " << reservationList[ctr].address<< "\n";

output << "preferable hairstyle: " << reservationList[ctr].preferable << "\n";

output << "hair stylist: "<<reservationList[ctr].hairstylist <<"\n";

output<< "-------------------Done----------------------\n";

ListSize++;

}

output.close();

break;

}

cout<<"File Exist, Data uploaded to program\n";

cout<< "-------------------Done----------------------\n";

input.close();

}

}

}

char Menu();

int main(){

char choice;

string name;

int counterCustomer = 0;

int counterHairstylist = 0;

int countreservation = 0;

// the control of the program.

do

{

choice=Menu();

switch(choice)

{

case '1':

cout<<"[1]\tHair Stylist\n";

cout<<"[2]\tCustomer\n\n";

cout<<"choice: ";

char c;

cin>>c;

if(c=='1'){

addHairstylist(counterHairstylist); // call function addHairstylist.

counterHairstylist++;

}

else if(c=='2'){

addCustomer(counterCustomer); // call function addCustomer.

counterCustomer++;

}

else{

cout<<"\tWrong Choice. try again...\n";

}

break;

case '2':

{

int x;

addreservation(countreservation); // call function addreservation.

countreservation++;

displayHairstylist(counterHairstylist); // call function displayHairstylist.

cout<<"Select Hairstylist: (enter her number)";

cin>>x;

FileLoad(countreservation,x); // call function FileLoad.

displayreservation(countreservation); // call function displayreservation.

}

break;

case '3':

{

cancelReservation(countreservation); // call function cancelReservation.

countreservation--;

displayreservation(countreservation); // call function displayreservation.

break;

}

break;

case '4':

cout<<"[1]\tHair Stylist\n";

cout<<"[2]\tCustomer\n\n";

cout<<"choice: ";

char c1;

cin>>c1;

if(c1=='1'){

cout<<"Enter the old first name: ";

cin>>name;

updatehairstylist(counterHairstylist); // call function updatehairstylist.

}

else if(c1=='2'){

cout<<"Enter the old first name: ";

cin>>name;

updatecustomer(counterCustomer); // call function updatecustomer.

}

else{

cout<<"\nWrong Choice. try again...\n";

}

break;

case '5':

cout<<"[1]\tHair Stylist\n";

cout<<"[2]\tCustomer\n\n";

cout<<"choice: ";

char c2;

cin>>c2;

if(c2=='1'){

cout<<"Enter your first Name: ";

string name;

cin>>name;

deleteHairstylist(name,counterHairstylist); // call function deleteHairstylist.

counterHairstylist--;

}

else if(c2=='2'){

cout<<"Enter your first Name: ";

string name2;

cin>>name2;

deleteCustomer(name2,counterCustomer); // call function deleteCustomer .

counterCustomer--;

}

else

cout<<"\tWrong Choice. try again...\n";

break;

case '6':

cout<<"[1]\tHair Stylist\n";

cout<<"[2]\tCustomer\n\n";

cout<<"choice: ";

char d;

cin>>d;

if(d=='1'){

displayHairstylist(counterHairstylist); // call function displayHairstylist .

}

else if(d=='2'){

displaycustomer(counterCustomer); // call function displayCustomer .

}

else{

cout<<"\tWrong Choice. try again...\n";

}

break;

case '7':

cout<<"Enter a First Name: "; // user will enter name to search about Customer .

cin>>name;

searchCustomer(name,counterCustomer ); // call function searchCustomer .

break;

case '8':

cout<<"Enter a First Name: "; // user will enter name to search about Hairstyl .

cin>>name;

searchHairstylist(name,counterHairstylist ); // call function searchHairstylist .

break;

case '0':cout<<"End of Program, Thank you\n"; // exit of the program.

break;

default :

cout<<"Wrong Choice Entry, Try Again\n";

}

}

while(choice!='0');

return 0;

}

// menu of the system features.

char Menu()

{

char ch;

cout<<"[1]\tAdd Custormer/Hairstylist\n";

cout<<"[2]\tNew Reservation\n";

cout<<"[3]\tCancel Reservation\n";

cout<<"[4]\tUpdate Customer/Hair Stylist details\n";

cout<<"[5]\tDelete Customer/Hair Stylist records\n";

cout<<"[6]\tShow Customers/Hair Stylist List\n";

cout<<"[7]\tSearch for specific customer details.(for hairStyle)\n";

cout<<"[8]\tSearch for available HairStylist.\n";

cout<<"[0]\tExit\n\n";

cout<<"Choice: ";

cin>>ch;

return ch;

}