CODE

#include <iostream>

#include <iomanip>

#include <fstream>

using namespace std;

// Caleb Sett 0706854811

// this Class represents node

class Node {

public:

int Room\_Number;

string Name;

string Address;

int Phone\_Number;

Node \*Next;

};

Node \*head = NULL;

ofstream record;

void Add\_Customer() {

Node \*temp = new Node();

// Asking the user to enter room number

cout << "Enter room number: ";

cin >> temp->Room\_Number;

// Asking the user to enter customer's name

cout << "Enter customer name: ";

cin.ignore();

getline(cin, temp->Name);

// prompting the user to enter address

cout << "Enter address: ";

getline(cin, temp->Address);

// request the user to enter phone number

cout << "Enter phone number: ";

cin >> temp->Phone\_Number;

temp->Next = NULL;

// New node will be a head node If the linked list is empty,

if (head == NULL) {

head = temp;

return;

}

// Finding the last node

Node \*last = head;

while (last->Next != NULL) {

last = last->Next;

}

// Inserting new node after last node (at the end)

last->Next = temp;

cout << endl;

}

void Display\_Record() {

Node \*temp = head;

record.open("record.txt", ofstream::out | ofstream::trunc);

// Header

cout << left << setw(20) << "Room No."

<< left << setw(25) << "Customer Name"

<< left << setw(20) << "Address"

<< left << setw(20) << "Phone Number" << endl;

cout << "-----------------------------------------------------------------------------" << endl;

// this will Loop over all nodes

while (temp != NULL) {

// Display node data

cout << left << setw(20) << temp->Room\_Number

<< left << setw(25) << temp->Name

<< left << setw(20) << temp->Address

<< left << setw(20) << temp->Phone\_Number << endl;

// Write records to the file

record << "Room Numnber: " << temp->Room\_Number << endl;

record << "Customer Name: " << temp->Name << endl;

record << "Address: " << temp->Address << endl;

record << "Phone Number: " << temp->Phone\_Number << endl;

record << endl;

temp = temp->Next;

}

record.close();

}

void Display\_Specific\_Record() {

int number;

Node \*temp = head;

// Asking user to enter room number for specific record

cout << "Enter Room Number: ";

cin >> number;

cout << endl;

while (temp != NULL) {

if ((temp->Room\_Number) == number) {

// Displaying the node data

cout << "Room Numnber: " << temp->Room\_Number << endl;

cout << "Customer Name: " << temp->Name << endl;

cout << "Address: " << temp->Address << endl;

cout << "Phone Number: " << temp->Phone\_Number << endl;

return;

}

temp = temp->Next;

}

cout << "Room is not booked by customer";

}

void Modify\_Record() {

int number, ch;

Node \*temp = head;

// Asking user to enter room number for data modification

cout << "Enter Room Number: ";

cin >> number;

cout << endl;

while (temp != NULL) {

if ((temp->Room\_Number) == number) {

// Ask user what do modify

cout << "What do you want to modify?\n";

cout << "1. Customer Name\n";

cout << "2. Customer Address\n";

cout << "3. Customer Phone Number\n";

cout << "Enter an option (1-3): ";

cin >> ch;

cin.ignore();

// Modify Record as user choice

switch(ch) {

case 1:

cout << "Enter New Name: ";

getline(cin, temp->Name);

break;

case 2:

cout << "Enter New Address: ";

getline(cin, temp->Address);

break;

case 3:

cout << "Enter New Phone Number: ";

cin >> temp->Phone\_Number;

break;

default: cout << "\nWrong Option!!!\n"; break;

}

return;

}

temp = temp->Next;

}

cout << "Record Not Found !!!";

}

void Delete\_Record() {

int number;

Node \*temp = head;

// Ask user to enter room number to delete

cout << "Enter Room Number: ";

cin >> number;

int pos = 0;

int flag = 0;

// Get position of node to delete

while (temp != NULL) {

pos++;

if ((temp->Room\_Number) == number) {

flag = 1;

break;

}

temp = temp->Next;

}

if(flag == 0) {

cout << "Room is not booked by customer";

} else {

// Delete the node at found position

Node \*tempx = head;

// If node to be deleted is head node

if (pos == 1) {

head = tempx->Next;

free(tempx);

return;

}

// Otherwise

for (int i = 2; tempx != NULL && i < pos - 1; i++) {

tempx = tempx->Next;

}

if (tempx == NULL || tempx->Next == NULL) {

return;

}

Node \*next = tempx->Next->Next;

free(tempx->Next); // Free node

tempx->Next = next;

cout << "\nRecord Deleted" << endl << endl;

}

}

int main() {

int choice;

do {

// Display menu

cout << "\n==============================================\n";

cout << right << setw(35) << "HOTEL RESERVATION SYSTEM\n";

cout << "==============================================\n";

cout << "\n\tPress 1 to BOOK A ROOM";

cout << "\n\tPress 2 to DISPLAY ALL ROOMS ALLOTTED";

cout << "\n\tPress 3 to DISPLAY SPECIFIC CUSTOMER RECORD";

cout << "\n\tPress 4 to MODIFY CUSTOMER RECORD";

cout << "\n\tPress 5 to DELETE CUSTOMER RECORD";

cout << "\n\tPress 6 to Exit";

cout << "\n\tEnter your choice: ";

cin >> choice;

cout << "\n";

switch (choice) {

case 1:

Add\_Customer();

break;

//function to display the record(s)

case 2:

Display\_Record();

break;

//displaying the specific record

case 3:

Display\_Specific\_Record();

break;

//a function to modify the record(s)

case 4:

Modify\_Record();

break;

//a function to delete the record(s)

case 5:

Delete\_Record();

break;

case 6:

break;

}

} while (choice != 6);

return 0;

}