***Assigment 04***

***Task 01***

#include <iostream>

#include <string>

using namespace std;

struct Book {

int id;

string title;

string author;

int quantity;

};

void addBook(Book books[], int& size)

{

cout << "Enter Book ID: ";

cin >> books[size].id;

cout << "Enter Book Title: ";

cin.ignore();

getline(cin, books[size].title);

cout << "Enter Book Author: ";

getline(cin, books[size].author);

cout << "Enter Book Quantity: ";

cin >> books[size].quantity;

size++;

}

void searchBook(Book books[], int size)

{

int choice;

cout << "Search by (1) ID or (2) Title: ";

cin >> choice;

if (choice == 1)

{

int id;

cout << "Enter Book ID: ";

cin >> id;

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

{

if (books[i].id == id) {

cout << "Book Found!" << endl;

cout << "Title: " << books[i].title << endl;

cout << "Author: " << books[i].author << endl;

cout << "Quantity: " << books[i].quantity << endl;

return;

}

}

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

} else if

(choice == 2)

{

string title;

cout << "Enter Book Title: ";

cin.ignore();

getline(cin, title);

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

{

if (books[i].title == title)

{

cout << "Book Found!" << endl;

cout << "ID: " << books[i].id << endl;

cout << "Author: " << books[i].author << endl;

cout << "Quantity: " << books[i].quantity << endl;

return;

}

}

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

}

}

void displayBooks(Book books[], int size)

{

cout << "Available Books:" << endl;

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

{

cout << "ID: " << books[i].id << endl;

cout << "Title: " << books[i].title << endl;

cout << "Author: " << books[i].author << endl;

cout << "Quantity: " << books[i].quantity << endl << endl;

}

}

int main()

{

const int MAX\_BOOKS = 100;

Book books[MAX\_BOOKS];

int size = 0;

while (true)

{

cout << "Library Management System" << endl;

cout << "1. Add a new book" << endl;

cout << "2. Search for a book" << endl;

cout << "3. Display all available books" << endl;

cout << "4. Exit" << endl;

cout << "Enter your choice: ";\

int choice;

cin >> choice;

switch (choice)

{

case 1:

if (size < MAX\_BOOKS)

addBook(books, size);

else

cout << "Maximum number of books reached!" << endl;

break;

case 2:

searchBook(books, size);

break;

case 3:

displayBooks(books, size);

break;

case 4:

return 0;

default:

cout << "Invalid choice! Please try again." << endl;

}

}

return 0;

}

***Task 02***

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Patient

{

int id;

string name;

int age;

string disease;

int roomNumber;

};

void addPatient(Patient patients[], int& size)

{

cout << "Enter Patient ID: ";

cin >> patients[size].id;

cout << "Enter Patient Name: ";

cin.ignore();

getline(cin, patients[size].name);

cout << "Enter Patient Age: ";

cin >> patients[size].age;

cout << "Enter Patient Disease: ";

cin.ignore();

getline(cin, patients[size].disease);

cout << "Enter Patient Room Number: ";

cin >> patients[size].roomNumber;

size++;

}

void searchPatient(Patient patients[], int size)

{

int id;

cout << "Enter Patient ID to search: ";

cin >> id;

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

{

if (patients[i].id == id)

{

cout << "Patient Found!" << endl;

cout << "Name: " << patients[i].name << endl;

cout << "Age: " << patients[i].age << endl;

cout << "Disease: " << patients[i].disease << endl;

cout << "Room Number: " << patients[i].roomNumber << endl;

return;

}

}

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

}

void displayPatients(Patient patients[], int size)

{

cout << "List of Patients:" << endl;

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

{

cout << "ID: " << patients[i].id << endl;

cout << "Name: " << patients[i].name << endl;

cout << "Age: " << patients[i].age << endl;

cout << "Disease: " << patients[i].disease << endl;

cout << "Room Number: " << patients[i].roomNumber << endl << endl;

}

}

void savePatientsToFile(Patient patients[], int size)

{

ofstream file("patients.txt");

if (file.is\_open())

{

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

{

file << patients[i].id << "," << patients[i].name << "," << patients[i].age << "," << patients[i].disease << "," << patients[i].roomNumber << endl;

}

file.close();

cout << "Patients saved to file successfully!" << endl;

}

else

{

cout << "Unable to open file!" << endl;

}

}

void loadPatientsFromFile(Patient patients[], int& size)

{

ifstream file("patients.txt");

if (file.is\_open())

{

string line;

while (getline(file, line))

{

stringstream ss(line);

string token;

getline(ss, token, ',');

patients[size].id = stoi(token);

getline(ss, token, ',');

patients[size].name = token;

getline(ss, token, ',');

patients[size].age = stoi(token);

getline(ss, token, ',');

patients[size].disease = token;

getline(ss, token, ',');

patients[size].roomNumber = stoi(token);

size++;

}

file.close();

cout << "Patients loaded from file successfully!" << endl;

}

else

{

cout << "Unable to open file!" << endl;

}

}

int main()

{

const int MAX\_PATIENTS = 100;

Patient patients[MAX\_PATIENTS];

int size = 0;

loadPatientsFromFile(patients, size);

while (true)

{

cout << "Hospital Patient Management System" << endl;

cout << "1. Add a new patient" << endl;

cout << "2. Search for a patient" << endl;

cout << "3. Display all patients" << endl;

cout << "4. Save and exit" << endl;

cout << "Enter your choice: ";

int choice;

cin >> choice;

switch (choice)

{

case 1:

if (size < MAX\_PATIENTS)

addPatient(patients, size);

else

cout << "Maximum number of patients reached!" << endl;

break;

case 2:

searchPatient(patients, size);

break;

case 3:

displayPatients(patients, size);

break;

case 4:

savePatientsToFile(patients, size);

return 0;

default: