****

**Submitted by: Raja Muhmmad Abdul Hadi**

**Roll no: 2430-0196**

**Class: BS-CS**

**ASSIGMENT NUMBER 4**

TASK NUMBER 1

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

int bookID;

string title;

string author;

int quantity;

};

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

Book newBook;

cout << "Enter Book ID: ";

cin >> newBook.bookID;

cout << "Enter Book Title: ";

cin.ignore();

getline(cin, newBook.title);

cout << "Enter Book Author: ";

getline(cin, newBook.author);

cout << "Enter Book Quantity: ";

cin >> newBook.quantity;

books[size] = newBook;

size++;

}

void searchBook(Book books[], int size) {

int choice;

int bookID;

string title;

cout << "Search by: " << endl;

cout << "1. Book ID" << endl;

cout << "2. Book Title" << endl;

cin >> choice;

if (choice == 1) {

cout << "Enter Book ID: ";

cin >> bookID;

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

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

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

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

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

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

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

return;

}

}

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

} else if (choice == 2) {

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 << "Book ID: " << books[i].bookID << endl;

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

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

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

return;

}

}

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

}

}

void displayBooks(Book books[], int size) {

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

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

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

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

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

cout << endl;

}

}

void saveInventory(Book books[], int size) {

ofstream file("inventory.txt");

if (file.is\_open()) {

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

file << books[i].bookID << endl;

file << books[i].title << endl;

file << books[i].author << endl;

file << books[i].quantity << endl;

}

file.close();

cout << "Inventory saved!" << endl;

} else {

cout << "Unable to save inventory!" << endl;

}

}

void loadInventory(Book books[], int &size) {

ifstream file("inventory.txt");

if (file.is\_open()) {

while (!file.eof()) {

file >> books[size].bookID;

file.ignore();

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

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

file >> books[size].quantity;

size++;

}

file.close();

cout << "Inventory loaded!" << endl;

} else {

cout << "Unable to load inventory!" << endl;

}

}

int main() {

const int MAX\_BOOKS = 100;

Book books[MAX\_BOOKS];

int size = 0;

loadInventory(books, size);

int choice;

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 books" << endl;

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

cin >> choice

switch (option) {

case 1:

addBook(books, size);

break;

case 2:

searchBook(books, size);

break;

case 3:

displayBooks(books, size);

break;

case 4:

cout << "Exiting..." << endl;

break;

default:

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

}

} while (option != 4);

return 0;

}

**TASK NUMBER 2**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct Patient {

int patientID;

string name;

int age;

string disease;

int roomNumber;

};

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

Patient newPatient;

cout << "Enter Patient ID: ";

cin >> newPatient.patientID;

cout << "Enter Patient Name: ";

cin.ignore();

getline(cin, newPatient.name);

cout << "Enter Patient Age: ";

cin >> newPatient.age;

cout << "Enter Patient Disease: ";

cin.ignore();

getline(cin, newPatient.disease);

cout << "Enter Room Number: ";

cin >> newPatient.roomNumber;

patients[size] = newPatient;

size++;

}

void searchPatient(Patient patients[], int size) {

int patientID;

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

cin >> patientID;

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

if (patients[i].patientID == patientID) {

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

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

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

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

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

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

return;

}

}

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

}

void displayPatients(Patient patients[], int size) {

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

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

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

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

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

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

cout << endl;

}

}

void savePatients(Patient patients[], int size) {

ofstream file("patients.txt");

if (file.is\_open()) {

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

file << patients[i].patientID << endl;

file << patients[i].name << endl;

file << patients[i].age << endl;

file << patients[i].disease << endl;

file << patients[i].roomNumber << endl;

}

file.close();

cout << "Patient records saved!" << endl;

} else {

cout << "Unable to save patient records!" << endl;

}

}

void loadPatients(Patient patients[], int &size) {

ifstream file("patients.txt");

if (file.is\_open()) {

while (!file.eof()) {

file >> patients[size].patientID;

file.ignore();

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

file >> patients[size].age;

file.ignore();

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

file >> patients[size].roomNumber;

size++;

}

file.close();

cout << "Patient records loaded!" << endl;

} else {

cout << "Unable to load patient records!" << endl;

}

}

int main() {

const int MAX\_PATIENTS = 100;

Patient patients[MAX\_PATIENTS];

int size = 0;

loadPatients(patients, size);

int choice;

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;

cin >> choice;

switch (choice) {

case 1:

addPatient(patients, size);

break;

case 2:

searchPatient(patients, size);

break;

case 3:

displayPatients(patients, size);

break;

case 4:

savePatients(patients, size);

return 0;

default:

cout << "Invalid choice!" << endl;

}

}

return 0;

}