

**Final project**



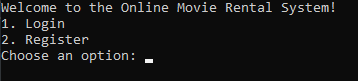
Online Movie Rental System Code

**PROGRAM:** Write a program to make a online rental system.

**Code#**  
#include <iostream>   
#include <fstream>   
#include <vector>   
#include <string>  
#include <sstream>  
#include <algorithm>  
#include <cctype>  
  
using namespace std;  
  
struct Movie {  
 string name;  
 int year;  
 double rent;  
};  
  
struct User {  
 string username;  
 string password;  
};  
  
vector<Movie> loadMovies() {  
 vector<Movie> movies;  
 ifstream movieFile("movies.txt");  
 string line;  
  
 while (getline(movieFile, line)) {  
 stringstream ss(line);  
 Movie movie;  
 ss >> ws; // eat whitespace  
 getline(ss, movie.name, ',');  
 ss >> movie.year;  
 ss >> movie.rent;  
 movies.push\_back(movie);  
 }  
 return movies;  
}  
  
vector<User> loadUsers() {  
 vector<User> users;  
 ifstream userFile("users.txt");  
 string line;  
  
 while (getline(userFile, line)) {  
 stringstream ss(line);  
 User user;  
 ss >> user.username >> user.password;  
 users.push\_back(user);  
 }  
 return users;  
}  
  
bool registerUser (const string& username, const string& password) {  
 ofstream userFile("users.txt", ios::app);  
 if (userFile.is\_open()) {  
 userFile << username << " " << password << endl;  
 userFile.close();  
 return true;  
 }  
 return false;  
}  
  
bool loginUser (const string& username, const string& password, const vector<User>& users) {  
 for (const auto& user : users) {  
 if (user.username == username && user.password == password) {  
 return true;  
 }  
 }  
 return false;  
}  
  
void displayMovies(const vector<Movie>& movies) {  
 cout << "Available Movies:  
";  
 for (size\_t i = 0; i < movies.size(); ++i) {  
 cout << i + 1 << ". Name: " << movies[i].name << ", Year: " << movies[i].year << ", Rent: $" << movies[i].rent << endl;  
 }  
}  
  
string toLower(const string& str) {  
 string lowerStr = str;  
 transform(lowerStr.begin(), lowerStr.end(), lowerStr.begin(), [](unsigned char c) { return tolower(c); });  
 return lowerStr;  
}  
  
void rentMovie(const string& username, vector<Movie>& movies, vector<string>& rentedMovies) {  
 displayMovies(movies);  
 string movieName;  
 cout << "Enter the name of the movie to rent: ";  
 cin.ignore();  
 getline(cin, movieName);  
 movieName = toLower(movieName); // Convert user input to lowercase  
  
 // Check if the movie exists  
 auto it = find\_if(movies.begin(), movies.end(), [&movieName](const Movie& movie) {  
 return toLower(movie.name) == movieName; // Convert movie name to lowercase for comparison  
 });  
  
 if (it != movies.end()) {  
 rentedMovies.push\_back(it->name); // Store the original movie name  
 cout << "You have rented: " << it->name << endl;  
  
 // Save rented movie to user's file  
 ofstream userRentals(username + "\_rentals.txt", ios::app);  
 if (userRentals.is\_open()) {  
 userRentals << it->name << endl; // Store the original movie name  
 userRentals.close();  
 }  
 } else {  
 cout << "Movie not found.  
";  
 }  
}  
  
void returnMovie(const string& username, vector<string>& rentedMovies) {  
 if (rentedMovies.empty()) {  
 cout << "You have no rented movies to return.  
";  
 return;  
 }  
  
 cout << "Your rented movies:  
";  
 for (size\_t i = 0; i < rentedMovies.size(); ++i) {  
 cout << i + 1 << ". " << rentedMovies[i] << endl;  
 }  
  
 int movieIndex;  
 cout << "Enter the number of the movie to return: ";  
 cin >> movieIndex;  
  
 if (movieIndex > 0 && movieIndex <= rentedMovies.size()) {  
 cout << "You have returned: " << rentedMovies[movieIndex - 1] << endl;  
 rentedMovies.erase(rentedMovies.begin() + (movieIndex - 1));  
  
 // Update user's rental file  
 ofstream userRentals(username + "\_rentals.txt");  
 if (userRentals.is\_open()) {  
 for (const auto& movie : rentedMovies) {  
 userRentals << movie << endl;  
 }  
 userRentals.close();  
 }  
 } else {  
 cout << "Invalid selection.  
";  
 }  
}  
  
vector<string> loadRentedMovies(const string& username) {  
 vector<string> rentedMovies;  
 ifstream userRentals(username + "\_rentals.txt");  
 string line;  
  
 while (getline(userRentals, line)) {  
 rentedMovies.push\_back(line);  
 }  
 return rentedMovies;  
}  
  
int main() {  
 vector<Movie> movies = loadMovies();  
 vector<User> users = loadUsers();  
 string username, password;  
 int choice;  
  
 cout << "Welcome to the Online Movie Rental System!  
";  
 cout << "1. Login  
2. Register  
Choose an option: ";  
 cin >> choice;  
  
 if (choice == 1) {  
 cout << "Enter username: ";  
 cin >> username;  
 cout << "Enter password: ";  
 cin >> password;  
  
 if (!loginUser(username, password, users)) {  
 cout << "Invalid username or password. Exiting.  
";  
 return 0;  
 }  
 } else if (choice == 2) {  
 cout << "Enter a new username: ";  
 cin >> username;  
 cout << "Enter a new password: ";  
 cin >> password;  
  
 if (registerUser(username, password)) {  
 cout << "Registration successful! You can now log in.  
";  
 } else {  
 cout << "Registration failed. Exiting.  
";  
 return 0;  
 }  
 } else {  
 cout << "Invalid choice. Exiting.  
";  
 return 0;  
 }  
  
 // Load user's rented movies  
 vector<string> rentedMovies = loadRentedMovies(username);  
  
 while (true) {  
 cout << "  
Options:  
1. Display Movies  
2. Rent Movie  
3. Return Movie  
4. Exit  
Choose an option: ";  
 cin >> choice;  
  
 if (choice == 1) {  
 displayMovies(movies);  
 } else if (choice == 2) {  
 rentMovie(username, movies, rentedMovies);  
 } else if (choice == 3) {  
 returnMovie(username, rentedMovies);  
 } else if (choice == 4) {  
 cout << "Exiting the system. Thank you!  
";  
 break;  
 } else {  
 cout << "Invalid choice. Please try again.  
";  
 }  
 }  
  
 return 0;  
}

**Outputs:**

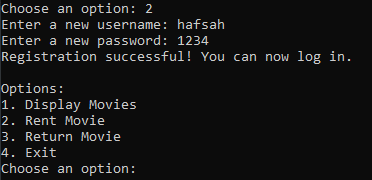
**Number1:**

****

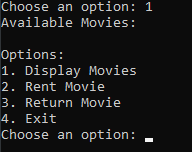
**Number2:**

****

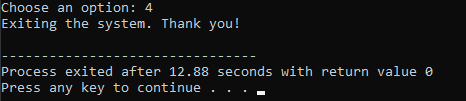
**Number3:**

****

**Number 4:**

****

**Number 5:**

****

**……………………❌………………❌…………………**

**The End**