1. #include <iostream>
2. #include <fstream>
3. #include <vector>
4. #include <string>
5. #include <algorithm>
6. #include <sstream>
7. #include <iomanip> // For setw and left
8. using namespace std;
9. struct Flight {
10. string flightNumber;
11. string destination;
12. int seatsAvailable;
13. };
14. struct Booking {
15. string username;
16. string flightNumber;
17. };
18. vector<Flight> flights;
19. vector<Booking> bookings;
20. // Function to load flights from a file
21. void loadFlights() {
22. ifstream file("flights.txt");
23. if (!file) {
24. cout << "No flight data found.\n";
25. return;
26. }
27. Flight f;
28. while (file >> f.flightNumber >> f.destination >> f.seatsAvailable) {
29. flights.push\_back(f);
30. }
31. file.close();
32. }
33. // Function to save flights to a file
34. void saveFlights() {
35. ofstream file("flights.txt");
36. if (!file) {
37. cerr << "Error: Unable to save flight data.\n";
38. return;
39. }
40. for (const auto& flight : flights) {
41. file << flight.flightNumber << " " << flight.destination << " " << flight.seatsAvailable << endl;
42. }
43. file.close();
44. }
45. // Function to load bookings from a file
46. void loadBookings() {
47. ifstream file("bookings.txt");
48. if (!file) {
49. return; // File might not exist yet
50. }
51. Booking b;
52. while (file >> b.username >> b.flightNumber) {
53. bookings.push\_back(b);
54. }
55. file.close();
56. }
57. // Function to save bookings to a file
58. void saveBookings() {
59. ofstream file("bookings.txt");
60. if (!file) {
61. cerr << "Error: Unable to save booking data.\n";
62. return;
63. }
64. for (const auto& booking : bookings) {
65. file << booking.username << " " << booking.flightNumber << endl;
66. }
67. file.close();
68. }
69. // Function to display flights in a formatted way
70. void displayFlights() {
71. cout << "\nAvailable Flights:\n";
72. cout << left << setw(15) << "Flight Number" << setw(15) << "Destination" << setw(15) << "Seats Available" << endl;
73. cout << string(45, '-') << endl; // Separator line
74. for (const auto& f : flights) {
75. cout << left << setw(15) << f.flightNumber << setw(15) << f.destination << setw(15) << f.seatsAvailable << endl;
76. }
77. }
78. // Function to book a flight
79. void bookFlight(string username) {
80. string flightNum;
81. cout << "Enter flight number to book: ";
82. cin >> flightNum;
83. for (auto& f : flights) {
84. if (f.flightNumber == flightNum && f.seatsAvailable > 0) {
85. f.seatsAvailable--;
86. bookings.push\_back({ username, flightNum });
87. cout << "Booking successful for " << username << " on Flight " << flightNum << "!\n";
88. saveFlights();
89. saveBookings();
90. return;
91. }
92. }
93. cout << "Flight not found or no seats available!\n";
94. }
95. // Function to display user bookings
96. void displayUserBookings(string username) {
97. cout << "\nYour Bookings:\n";
98. cout << left << setw(15) << "Flight Number" << endl;
99. cout << string(15, '-') << endl;
100. for (const auto& b : bookings) {
101. if (b.username == username) {
102. cout << left << setw(15) << b.flightNumber << endl;
103. }
104. }
105. }
106. //Function to cancel a booking.
107. void cancelBooking(string username){
108. string flightNum;
109. cout << "Enter flight number to cancel: ";
110. cin >> flightNum;
111. for(auto it = bookings.begin(); it != bookings.end(); ++it){
112. if(it->username == username && it->flightNumber == flightNum){
113. for(auto &f : flights){
114. if(f.flightNumber == flightNum){
115. f.seatsAvailable++;
116. break;
117. }
118. }
119. bookings.erase(it);
120. cout << "Booking for flight " << flightNum << " cancelled.\n";
121. saveFlights();
122. saveBookings();
123. return;
124. }
125. }
126. cout << "Booking not found.\n";
127. }
128. //Function to search flights by destination.
129. void searchFlightsByDestination(string destination){
130. cout << "\nFlights to " << destination << ":\n";
131. cout << left << setw(15) << "Flight Number" << setw(15) << "Seats Available" << endl;
132. cout << string(30, '-') << endl;
133. for(const auto &f : flights){
134. if(f.destination == destination){
135. cout << left << setw(15) << f.flightNumber << setw(15) << f.seatsAvailable << endl;
136. }
137. }
138. }
139. // Dummy login function
140. bool login(string& username) {
141. string password;
142. cout << "Enter username: ";
143. cin >> username;
144. cout << "Enter password: ";
145. cin >> password; // In a real system, use hashing
146. return true; // Dummy authentication
147. }
148. int main() {
149. loadFlights();
150. loadBookings();
151. string username;
152. if (login(username)) {
153. cout << "Login successful!\n";
154. int choice;
155. do {
156. cout << "\n1. Display Flights\n2. Book Flight\n3. Display Bookings\n4. Cancel Booking\n5. Search Flights by Destination\n0. Exit\nEnter choice: ";
157. cin >> choice;
158. switch (choice) {
159. case 1:
160. displayFlights();
161. break;
162. case 2:
163. bookFlight(username);
164. break;
165. case 3:
166. displayUserBookings(username);
167. break;
168. case 4:
169. cancelBooking(username);
170. break;
171. case 5: {
172. string destination;
173. cout << "Enter destination to search: ";
174. cin >> destination;
175. searchFlightsByDestination(destination);
176. break;
177. }
178. case 0:
179. cout << "Exiting...\n";
180. break;
181. default:
182. cout << "Invalid choice!\n";
183. }
184. } while (choice != 0);
185. } else {
186. cout << "Login failed!\n";
187. }
188. return 0;
189. }