**//============================================================================**

**// Name : Personal Dairy Management**

**// Author : UNKNOWN**

**// Version : V2**

**// Copyright : Your copyright notice**

**// Description : Hell Nah**

**//============================================================================**

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using namespace std;

// Structure for storing user information

struct User {

string username;

string password;

};

// Structure for storing diary entry

struct DiaryEntry {

string date;

string title;

string content;

};

// Function to display menu options

void displayMenu() {

cout << "\nDiary Management System\n";

cout << "1. Login\n";

cout << "2. Register\n";

cout << "3. Exit\n";

cout << "Enter your choice: ";

}

// Function to read user data from file

vector<User> readUserData() {

vector<User> users;

ifstream file("users.txt");

if (file.is\_open()) {

while (!file.eof()) {

User user;

getline(file, user.username);

getline(file, user.password);

if (!user.username.empty() && !user.password.empty()) {

users.push\_back(user);

}

}

file.close();

}

return users;

}

// Function to write user data to file

void writeUserData(vector<User> users) {

ofstream file("users.txt");

if (file.is\_open()) {

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

file << users[i].username << endl;

file << users[i].password << endl;

}

file.close();

}

}

// Function to authenticate user

bool authenticateUser(string username, string password, vector<User> users) {

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

if (users[i].username == username && users[i].password == password) {

return true;

}

}

return false;

}

// Function to display diary entry

void displayDiaryEntry(DiaryEntry entry) {

cout << "\nDate: " << entry.date << endl;

cout << "Title: " << entry.title << endl;

cout << "Content: " << entry.content << endl;

}

// Function to add diary entry

void addDiaryEntry(string username) {

cout << "\nEnter diary entry details:\n";

DiaryEntry entry;

cout << "Date (YYYY-MM-DD): ";

getline(cin, entry.date);

cout << "Title: ";

getline(cin, entry.title);

cout << "Content: ";

getline(cin, entry.content);

// Save entry to file

ofstream file(username + ".txt", ios\_base::app);

if (file.is\_open()) {

file << entry.date << endl;

file << entry.title << endl;

file << entry.content << endl;

file.close();

cout << "\nDiary entry added successfully!\n";

}

}

// Function to view all diary entries

void viewAllDiaryEntries(string username) {

ifstream file(username + ".txt");

if (file.is\_open()) {

DiaryEntry entry;

while (getline(file, entry.date)) {

getline(file, entry.title);

getline(file, entry.content);

displayDiaryEntry(entry);

}

file.close();

}

}

// Function to display user options

void displayUserOptions(string username) {

cout << "\nDiary Management System\n";

cout << "Logged in as: " << username << endl;

cout << "1. Add diary entry\n";

cout << "2. View all diary entries\n";

cout << "3. Logout\n";

cout << "Enter your choice: ";

}

int main() {

vector<User> users = readUserData();

int choice = 0;

string username, password;

bool authenticated = false;

while (choice != 3) {

displayMenu();

cin >> choice;

cin.ignore();

if (choice == 1) {

// Login

cout << "\nEnter username: ";

getline(cin, username);

cout << "Enter password: ";

getline(cin, password);

authenticated = authenticateUser(username, password, users);

if (authenticated) {

cout << "\nLogin successful!\n";

int userChoice = 0;

while (userChoice != 3) {

displayUserOptions(username);

cin >> userChoice;

cin.ignore();

if (userChoice == 1) {

// Add diary entry

addDiaryEntry(username);

} else if (userChoice == 2) {

// View all diary entries

viewAllDiaryEntries(username);

} else if (userChoice == 3) {

// Logout

authenticated = false;

cout << "\nLogout successful!\n";

} else {

cout << "\nInvalid choice! Please try again.\n";

}

}

} else {

cout << "\nInvalid username or password! Please try again.\n";

}

} else if (choice == 2) {

// Register

cout << "\nEnter username: ";

getline(cin, username);

cout << "Enter password: ";

getline(cin, password);

User newUser = {username, password};

users.push\_back(newUser);

writeUserData(users);

cout << "\nRegistration successful!\n";

} else if (choice == 3) {

// Exit

cout << "\nExiting program...\n";

} else {

cout << "\nInvalid choice! Please try again.\n";

}

}

    return 0;

}