
ITM SKILLS UNIVERSITY
itm
ITM GROUP OF INSTITUTIONS
ITM SKILLS UNIVERSITY

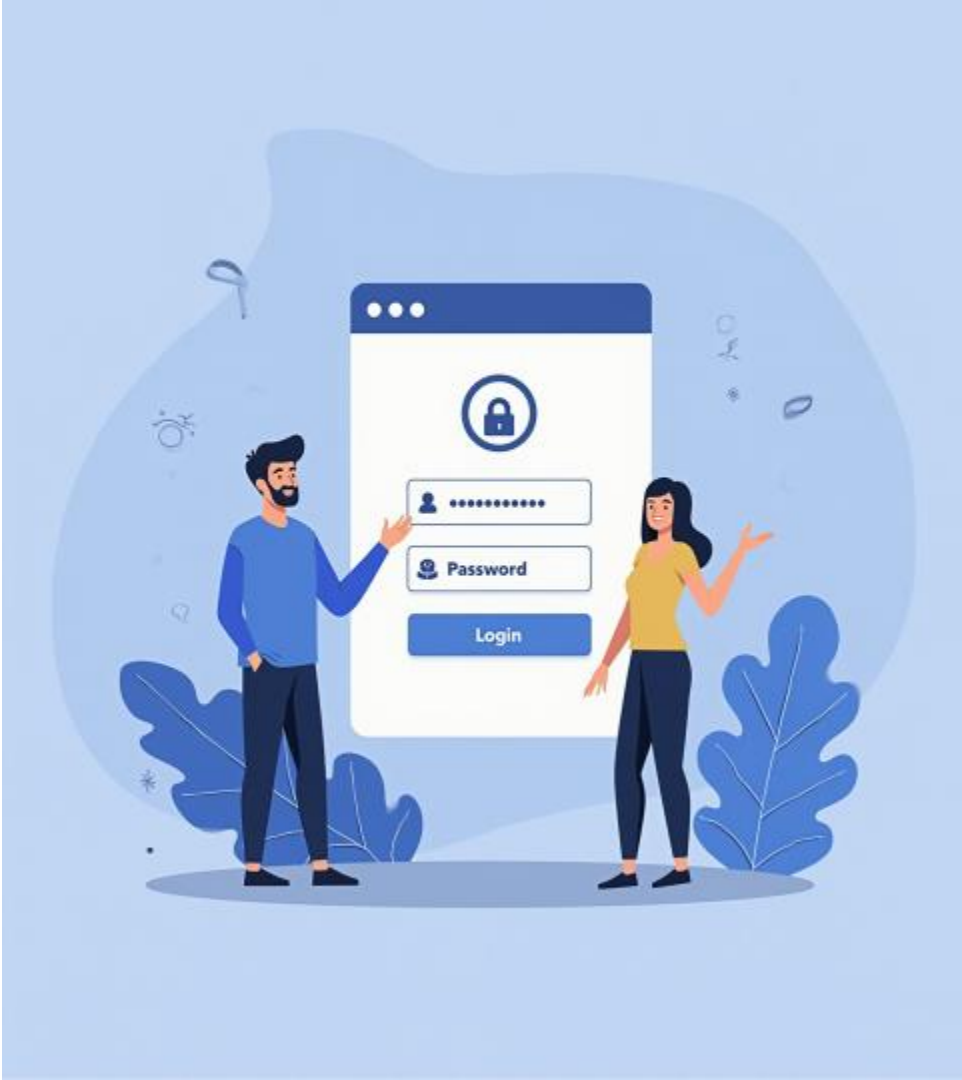
C++
Group-12

Introduction

This presentation explains a simple login system implemented in C++. It covers the core features such as user registration, login process, and password hashing. The goal is to create a secure, easy-to-use system with data stored in a plain text file. Visual aids like login form illustrations enhance understanding.

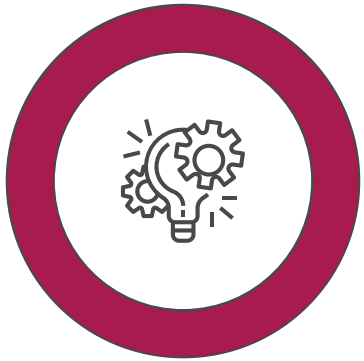
01

Introduction to Login System



Purpose and Features

The system enables users to register and log in securely by storing hashed passwords. Key features include simple user management, password hashing to protect credentials, and data storage in a text file named *users.txt*. This approach balances ease of implementation and basic security needs.



User Registration Process

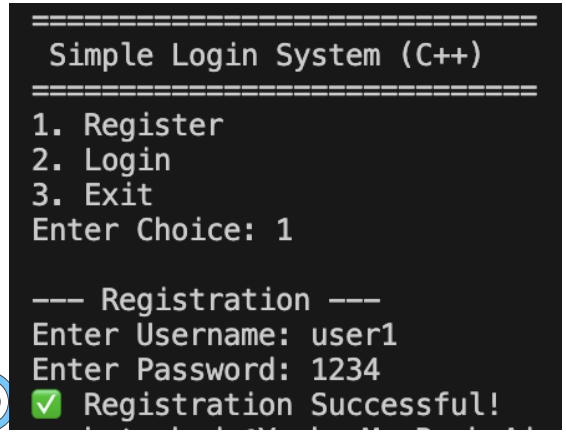
Users provide a username and password during registration. The password is hashed using a custom function before saving, ensuring it’s not stored in plaintext. Each entry is recorded in *users.txt*. This process guarantees that sensitive data is handled securely from the start of user interaction.

02 Code Implementation

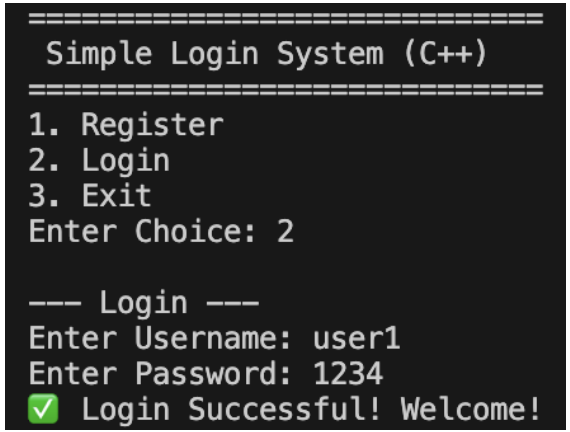
```cpp
#include <iostream>
#include <fstream>
#include <string>
#include <functional>

using namespace std;

string hashPassword(string password)
{
    hash<string> hasher;
    return to_string(hasher(password));
}

void registerUser()
{
    string username, password;

    cout << "\n--- Registration ---\n";
    cout << "Enter Username: ";
    cin >> username;

    cout << "Enter Password: ";
    cin >> password;

    ofstream file("users.txt", ios::app);

    if (file.is_open())
    {
        file << username << " " << hashPassword(password) << endl;
        file.close();
        cout << "✅ Registration Successful!\n";
    }
    else
    {
        cout << "❌ File Error!\n";
    }
}

bool loginUser()
{
    string username, password;
    string storedUser, storedPass;

    cout << "\n--- Login ---\n";
    cout << "Enter Username: ";
    cin >> username;

    cout << "Enter Password: ";
    cin >> password;

    string hashedPass = hashPassword(password);

    ifstream file("users.txt");

    if (file.is_open())
    {
        while (file >> storedUser >> storedPass)
        {
            if (storedUser == username && storedPass == hashedPass)
            {
                file.close();
                return true;
            }
        }
        file.close();
    }

    return false;
}

int main()
{
    int choice;

    cout << "=============================\n";
    cout << " Simple Login System (C++)\n";
    cout << "=============================\n";
    cout << "1. Register\n";
    cout << "2. Login\n";
    cout << "3. Exit\n";
    cout << "Enter Choice: ";

    // Input validation
    if (!(cin >> choice))
    {
        cout << "\n❌ Invalid Input! Please enter only number (1/2/3).\n";
        return 0;
    }

    if (choice == 1)
    {
        registerUser();
    }
    else if (choice == 2)
    {
        if (loginUser())
            cout << "✅ Login Successful! Welcome!\n";
        else
            cout << "❌ Invalid Username or Password!\n";
    }
    else if (choice == 3)
    {
        cout << "Exiting Program...\n";
    }
    else
    {
        cout << "❌ Invalid Choice! Enter only 1, 2 or 3.\n";
    }

    return 0;
}
```

Output

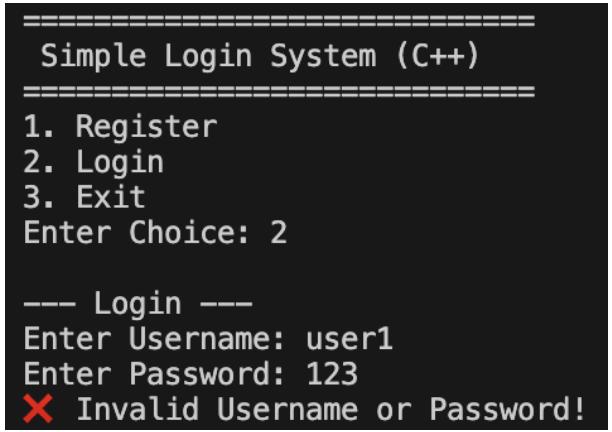
Register



Login



If user or password invalid



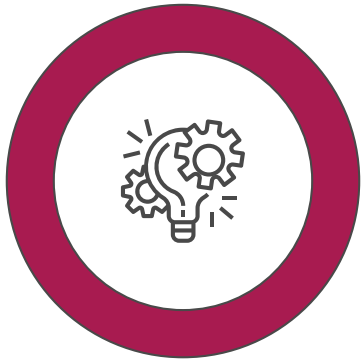
Output

Local User Data Storage (user.txt)

output > users.txt

```
1    user1 16197558426068326720
2    user2 15350671659626735038
3    user3 16129511837821584429
4    user4 4665927059433387458
5    user5 4340707622548420052
```

- User data is stored **locally** in a text file named The file is created in the **program output directory**
- C++ **file handling (**is used to store and read data
- Each line represents **one registered user**
- Data format used:
- Passwords are **hashed**, not stored in plain text
- Improves basic **data security**
- Suitable for **offline and small-scale login systems**
- No database or internet connection required



Conclusions

This system demonstrates a fundamental and **secure user login** process utilizing password hashing. Although basic, it effectively protects user data stored in plain text files. Future improvements could include adopting stronger hash algorithms and integrating database storage for enhanced scalability and security.


ITM SKILLS UNIVERSITY
itm
ITM GROUP OF INSTITUTIONS
ITM SKILLS UNIVERSITY

Thanks