

BSc. (Software Engineering)

Object Oriented Programming

Student: Md. Ashraful Islam Chowdhury

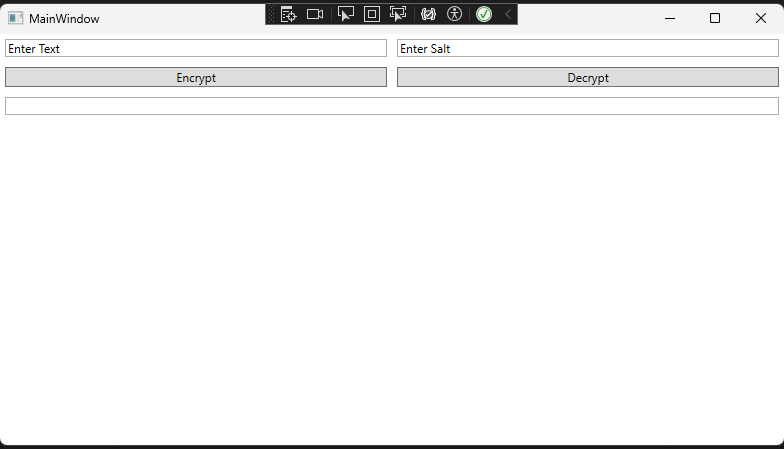
Report

The Password Reset Application serves as an example to illustrate password encryption and brute force decryption concepts using a multi-threaded method. It includes a WPF graphical interface where users can enter a password and salt, encrypt the password, and then carry out a brute force attack to decrypt it.

Github Link: **https://github.com/Ashraful-Islam101/C-final-task**

(Because of some difficulties I couldn’t upload the zip file to moodle learning environment)

Interface



**Device**

Hardware:   
1. Processor: Intel Core i5 11th Gen

2. RAM: 8GB

3. Storage: NVMe 512GB

Software:

1. Operating System: Windows 11   
2. Development Environment: Visual Studio 2022

3.. .NET Framework: .NET 8.0

**FUNCTIONAL REQUIREMENTS**

1. Encrypt Password Section

- Users can input a password and a salt.

- The application encrypts the password with the provided salt.

2. Decrypt Section

- The application attempts to decrypt the password using a brute force attack.

- The duration of the brute force attack is measured and displayed.

- The decrypted password and the time taken are displayed.

**TEST CASES**

1. User Interface Testing

- Test Case 1.1: Verify the presence of all UI elements (text boxes, buttons, dropdown, text blocks).

- Expected Result: All UI elements should be visible and correctly placed.

- Actual Result: Successful.

2. Encrypt Password Functionality

- Test Case 2.1: Encrypt a password with a given salt.

- Steps:

1. Enter a password in the "Enter text" text box.

2. Enter salt in the "Enter Salt" text box.

3. Click the " Encrypt " button.

- Expected Result: The password should be encrypted.

- Actual Result: Successful

- Test Case 2.2: Encrypt a password without entering a salt.

- Steps:

1. Enter a password in the "Enter text" text box.

2. Leave the "Enter Salt" text box empty.

3. Click the " Encrypt " button.

- Expected Result: Nothing is displayed.

- Actual Result: Successful. Nothing is displayed.

3. Decrypt Functionality

- Test Case 3.1: Perform decryption using brute force.

- Steps:

1. Encrypt a password using the steps from Test Case 2.1.

2. Click the "decrypt" button.

-Expected Result: The application should attempt to decrypt the password with brute force. The decrypted password and time taken should be displayed.

- Actual Result: Successful.

**Key Components and Their Descriptions**

1. App.xaml & App.xaml.cs:

- These files define the application-level settings and behaviors. `App.xaml` typically includes resource dictionaries, and `App.xaml.cs` contains the entry point and startup logic for the application.

2. MainWindow.xaml & MainWindow.xaml.cs:

- These files define the main window of the WPF application. `MainWindow.xaml` contains the XAML markup for the UI, and `MainWindow.xaml.cs` includes the code-behind logic that interacts with the UI elements.

3. Project Files (`.csproj` and `.sln`):

- `ASH\_Decrypt.csproj` is the project file that contains configurations and dependencies for building the project.

- `ASH\_Decrypt.sln` is the solution file that can include multiple projects but, in this case, appears to only include `ASH\_Decrypt`.

4. AssemblyInfo.cs:

- This file contains metadata about the assembly, such as version information and other attributes.