Semester Project Proposal

Information Security

Project Title

Secure Password Manager

Submitted to:

Ma’am Muniba Khan

Group Members:

- Mahnoor Malik **(UW-24-AI-BS-005)**
- Rida Hashim **(UW-24-AI-BS-012)**
- Menahil Rasheed **(UW-24-AI-BS-039)**

Session:

2024-2028

Table of Contents

1. Introduction

With the rapid growth of digital systems, password security has become a critical concern. Weak or reused passwords are one of the major causes of security breaches. Users often struggle to create strong passwords and to manage them securely. This project, **Secure Password Manager**, is developed to address these issues by providing a secure mechanism for user authentication, password strength evaluation, and secure password generation.

The project demonstrates core **Information Security concepts** such as password hashing, session management, CSRF protection, and audit logging using Python and Tkinter. It is designed as a desktop-based application to help users understand and apply basic security practices.

2. Features

The Secure Password Manager provides the following features:

- User Signup and Login system
- Secure password storage using SHA-256 hashing
- Password strength checker (Weak / Medium / Strong)
- Secure random password generator
- CSRF token validation for sensitive operations
- Session management using session tokens
- Audit logging of security-related actions
- Graphical User Interface (GUI) using Tkinter
- Logout functionality with token reset

3. Objective of the Project

The main objectives of this project are:

- To implement secure user authentication mechanisms
- To demonstrate password hashing instead of plain text storage
- To educate users about strong password creation
- To implement CSRF protection conceptually
- To log user activities for security auditing
- To provide hands-on understanding of Information Security concepts

4. Project Scope

This project can be used in the following areas:

- Academic learning and Information Security demonstrations
- Educational institutions for teaching password security concepts
- Small-scale desktop applications requiring basic authentication
- Prototype systems for understanding secure authentication flows

The project is intended as a **prototype** and learning tool, not a full commercial password manager.

5. Tools and Techniques

Tools Used

- Python 3
- Tkinter (GUI development)
- File handling (Text files)

Techniques Used

- SHA-256 password hashing
- Session token generation using UUID
- CSRF token validation
- Regular expressions for password strength checking
- Audit logging with timestamps

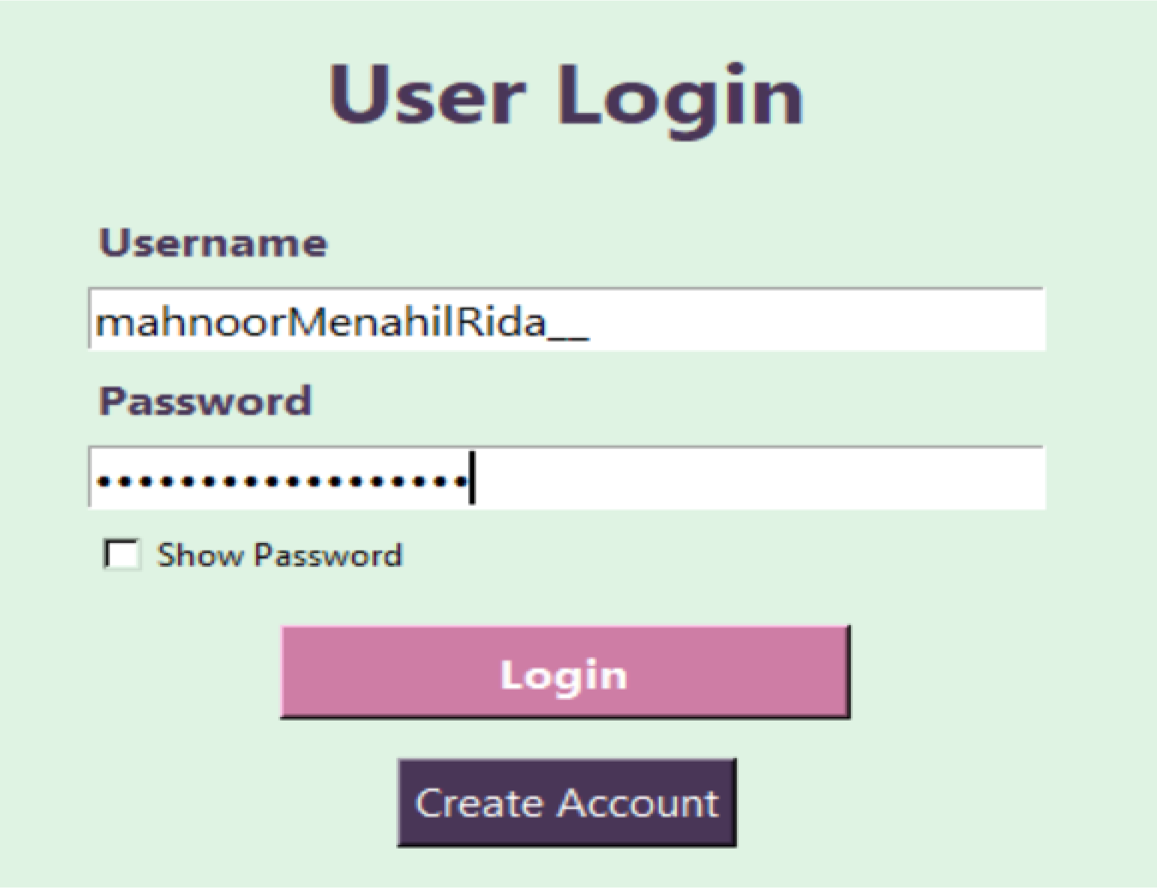
6. Implementation

The project is divided into two main parts:

6.1 User Authentication Module

- Handles user signup and login
- Passwords are hashed using SHA-256 before storage
- User data is stored in a text file (`users.txt`)

  Figure 1: User Login Screen



Username

mahnoorMenahilRida__

Password

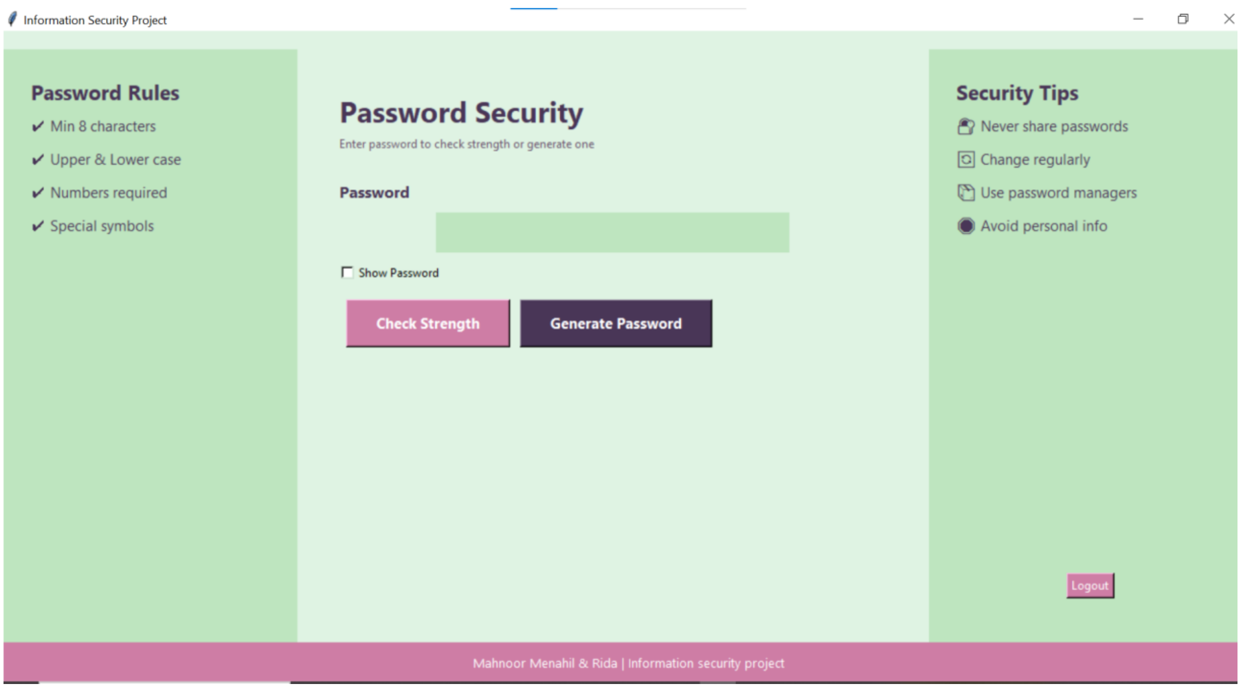
••••••••••••••••••

This screen allows existing users to log in using their credentials.

6.2 CSRF and Session Management

- CSRF tokens are generated after login
- Sensitive actions such as password strength checking and generation require CSRF token validation
- Session tokens identify active user sessions

Figure 2: Dashboard Screen

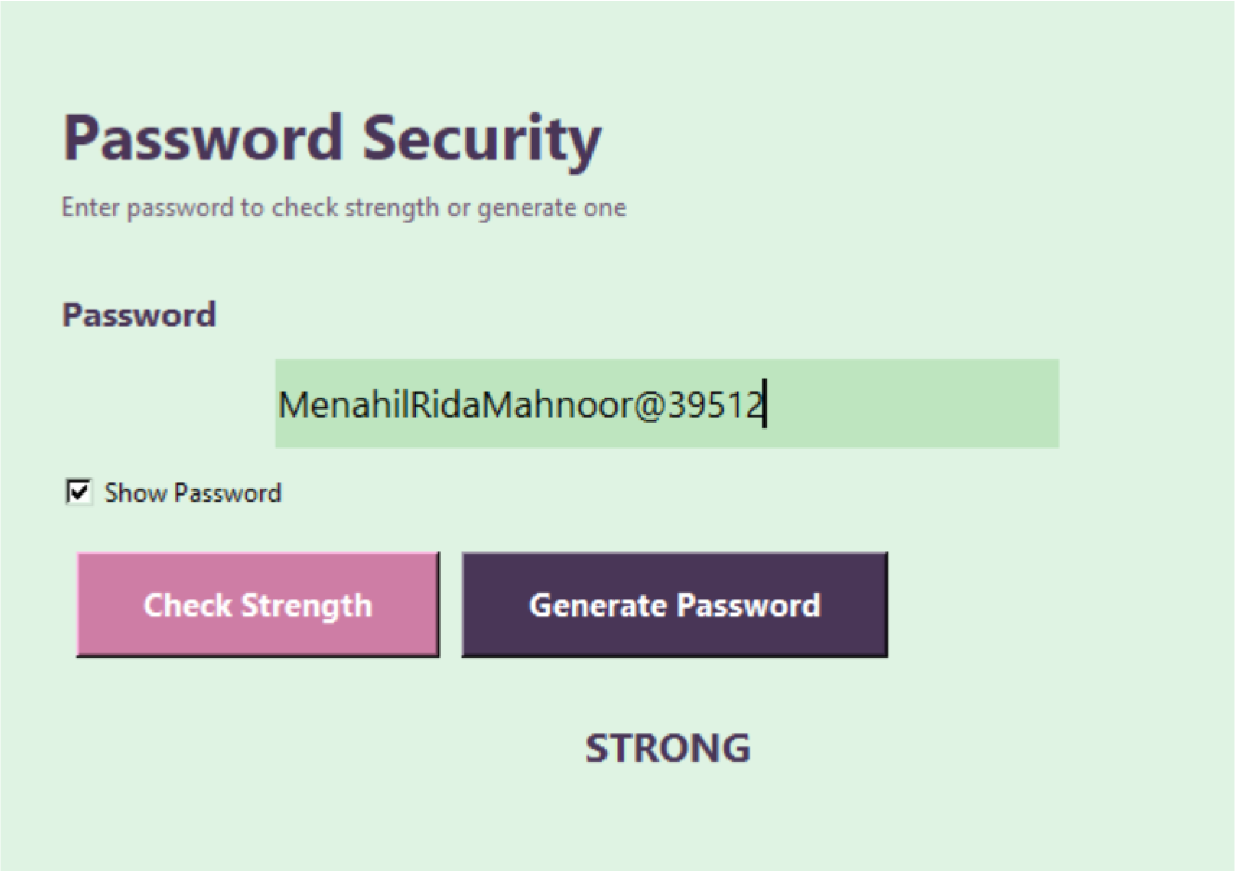


This screen represents the main dashboard after successful login.

6.3 Password Strength Checker

- Passwords are evaluated based on:
  - Length
  - Uppercase letters
  - Lowercase letters
  - Numbers
  - Special characters
- Password strength is classified as Weak, Medium, or Strong

Figure 3: Password Strength Checking



Password Security

Enter password to check strength or generate one

Password

MenahilRidaMahnoor@39512

☑ Show Password

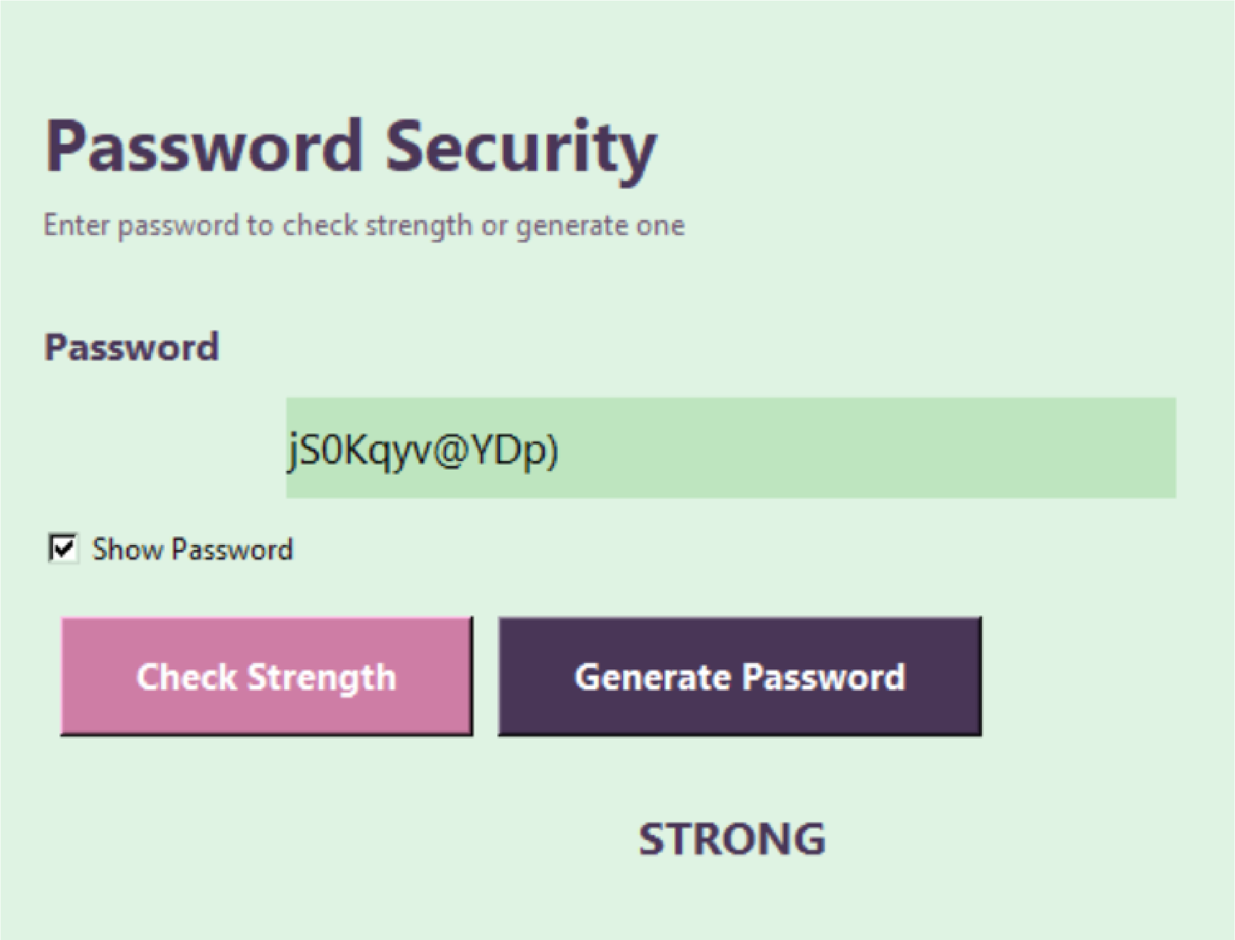
STRONG

This figure shows password strength evaluation result.

6.4 Secure Password Generator

- Generates a strong random password
- Ensures inclusion of:
    - Uppercase letters
    - Lowercase letters
    - Numbers
    - Special characters

Figure 4: Password Generation



Password Security

Enter password to check strength or generate one

Password: jS0Kqyv@YDp)

☑ Show Password

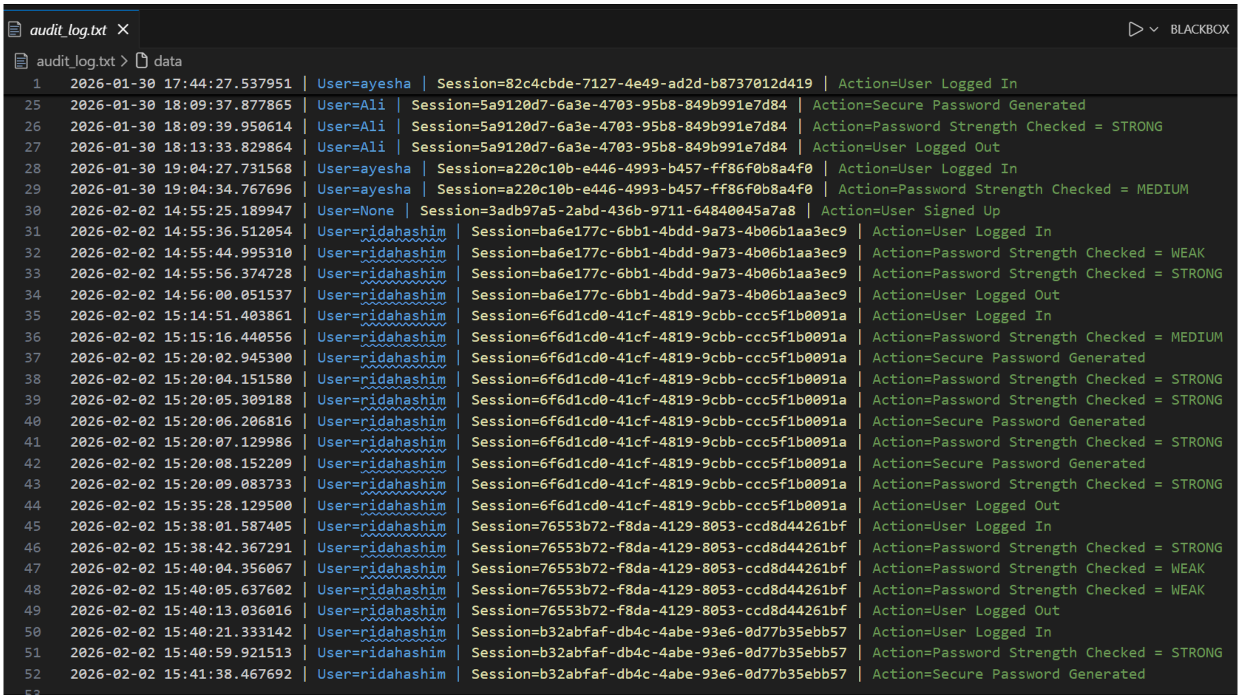
STRONG

This figure displays an automatically generated secure password.

6.5 Audit Logging

- All security-related actions are logged:
    - Login attempts
    - Password checks
    - Token misuse
    - Logout actions

Figure 5: Audit Log File

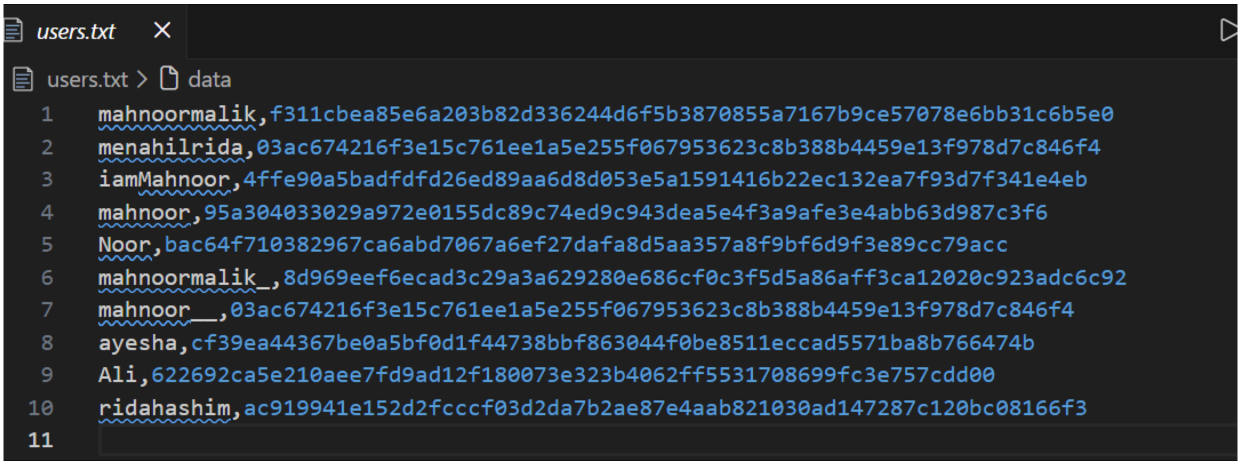


```
2026-01-30 17:44:27.537951 | User=ayesha | Session=82c4cbde-7127-4e49-ad2d-b8737012d419 | Action=User Logged In
2026-01-30 18:09:37.877865 | User=Ali | Session=5a9120d7-6a3e-4703-95b8-849b991e7d84 | Action=Secure Password Generated
2026-01-30 18:09:39.950614 | User=Ali | Session=5a9120d7-6a3e-4703-95b8-849b991e7d84 | Action=Password Strength Checked = STRONG
2026-01-30 18:13:33.829864 | User=Ali | Session=5a9120d7-6a3e-4703-95b8-849b991e7d84 | Action=User Logged Out
2026-01-30 19:04:27.731568 | User=ayesha | Session=a220c10b-e446-4993-b457-ff86f0b8a4f0 | Action=User Logged In
2026-01-30 19:04:34.767696 | User=ayesha | Session=a220c10b-e446-4993-b457-ff86f0b8a4f0 | Action=Password Strength Checked = MEDIUM
2026-02-02 14:55:25.189947 | User=None | Session=3adb97a5-2abd-436b-9711-64840045a7a8 | Action=User Signed Up
2026-02-02 14:55:36.512054 | User=ridahashim | Session=ba6e177c-6bb1-4bdd-9a73-4b06b1aa3ec9 | Action=User Logged In
2026-02-02 14:55:44.995310 | User=ridahashim | Session=ba6e177c-6bb1-4bdd-9a73-4b06b1aa3ec9 | Action=Password Strength Checked = WEAK
2026-02-02 14:55:56.374728 | User=ridahashim | Session=ba6e177c-6bb1-4bdd-9a73-4b06b1aa3ec9 | Action=Password Strength Checked = STRONG
2026-02-02 14:56:00.051537 | User=ridahashim | Session=ba6e177c-6bb1-4bdd-9a73-4b06b1aa3ec9 | Action=User Logged Out
2026-02-02 15:14:51.403861 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=User Logged In
2026-02-02 15:15:16.440556 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Password Strength Checked = MEDIUM
2026-02-02 15:20:02.945300 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Secure Password Generated
2026-02-02 15:20:04.151580 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Password Strength Checked = STRONG
2026-02-02 15:20:05.309188 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Password Strength Checked = STRONG
2026-02-02 15:20:06.206816 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Secure Password Generated
2026-02-02 15:20:07.129986 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Password Strength Checked = STRONG
2026-02-02 15:20:08.152209 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Secure Password Generated
2026-02-02 15:20:09.083733 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=Password Strength Checked = STRONG
2026-02-02 15:35:28.129500 | User=ridahashim | Session=6f6d1cd0-41cf-4819-9cbb-ccc5f1b0091a | Action=User Logged Out
2026-02-02 15:38:01.587405 | User=ridahashim | Session=76553b72-f8da-4129-8053-ccd8d44261bf | Action=User Logged In
2026-02-02 15:38:42.367291 | User=ridahashim | Session=76553b72-f8da-4129-8053-ccd8d44261bf | Action=Password Strength Checked = STRONG
2026-02-02 15:40:04.356067 | User=ridahashim | Session=76553b72-f8da-4129-8053-ccd8d44261bf | Action=Password Strength Checked = WEAK
2026-02-02 15:40:05.637602 | User=ridahashim | Session=76553b72-f8da-4129-8053-ccd8d44261bf | Action=Password Strength Checked = WEAK
2026-02-02 15:40:13.036016 | User=ridahashim | Session=76553b72-f8da-4129-8053-ccd8d44261bf | Action=User Logged Out
2026-02-02 15:40:21.333142 | User=ridahashim | Session=b32abfaf-db4c-4abe-93e6-0d77b35ebb57 | Action=User Logged In
2026-02-02 15:40:59.921513 | User=ridahashim | Session=b32abfaf-db4c-4abe-93e6-0d77b35ebb57 | Action=Password Strength Checked = STRONG
2026-02-02 15:41:38.467692 | User=ridahashim | Session=b32abfaf-db4c-4abe-93e6-0d77b35ebb57 | Action=Secure Password Generated
```

This figure shows recorded user activities in the audit log.

6.6 User Credentials Storage (users.txt)

Figure 6: User Credentials Stored in users.txt



```
mahnoormalik,f311cbea85e6a203b82d336244d6f5b3870855a7167b9ce57078e6bb31c6b5e0
menahilrida,03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4
iamMahnoor,4ffe90a5badfdfd26ed89aa6d8d053e5a1591416b22ec132ea7f93d7f341e4eb
mahnoor,95a304033029a972e0155dc89c74ed9c943dea5e4f3a9afe3e4abb63d987c3f6
Noor,bac64f710382967ca6abd7067a6ef27dafa8d5aa357a8f9bf6d9f3e89cc79acc
mahnoormalik_,8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3ca12020c923adc6c92
mahnoor__,03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4
ayesha,cf39ea44367be0a5bf0d1f44738bbf863044f0be8511eccad5571ba8b766474b
Ali,622692ca5e210aee7fd9ad12f180073e323b4062ff5531708699fc3e757cdd00
ridahashim,ac919941e152d2fcccf03d2da7b2ae87e4aab821030ad147287c120bc08166f3
```

This figure shows the contents of the users.txt file where usernames and passwords are stored in hashed (SHA-256) format instead of plain text.

7. Results

The project successfully achieves its objectives:

- Users can securely create and authenticate accounts
- Passwords are never stored in plain text
- Strong passwords are generated automatically

- Weak passwords are detected effectively
- CSRF tokens prevent unauthorized actions
- All actions are recorded for security auditing

The application runs smoothly with a user-friendly interface.

8. Conclusion

The Secure Password Manager project successfully demonstrates fundamental Information Security concepts through a practical implementation. By integrating hashing, CSRF protection, session management, and audit logging, the project provides a secure and educational system. Although it is a prototype, it effectively highlights the importance of password security and secure authentication mechanisms.

9. Future Works

The project can be enhanced in the future by:

- Using bcrypt or Argon2 instead of SHA-256
- Encrypting stored passwords using AES
- Storing data in a secure database instead of text files
- Implementing session expiration and timeout
- Adding multi-factor authentication (MFA)
- Protecting audit logs from tampering

10. References

1. Stallings, W. *Cryptography and Network Security*, Pearson
2. OWASP Foundation – Password Security Guidelines
3. Python Official Documentation
4. Tkinter Documentation
5. NIST Digital Identity Guidelines