SECURE PASSWORD MANAGER

Team Members:

Mahnoor Malik(UW-24-AI-BS-005)

Rida Hashim (UW-24-AI-BS-012)

Menahil Rasheed (UW-24-AI-BS-039)

Project Overview:

User sign up and login

Hashes passwords using SHA-256

Generates secure passwords

Checks password strength

CSRF token validation

Logs all actions in audit file

Problem Statement:

Weak passwords can be easily hacked using:

- Brute-force attacks
- Dictionary attacks
- Data breaches

Features:

- Sign up and login
- Secure Password generator
- Password strength checker
- SHA-256 hash display
- Show/Hide password
- Audit logging

Password Rules:

A strong password must include:

- **Minimum 8 characters**
- **Uppercase letter**
- **Lowercase letter**
- **Number**
- **Special symbol**

Strength Levels:

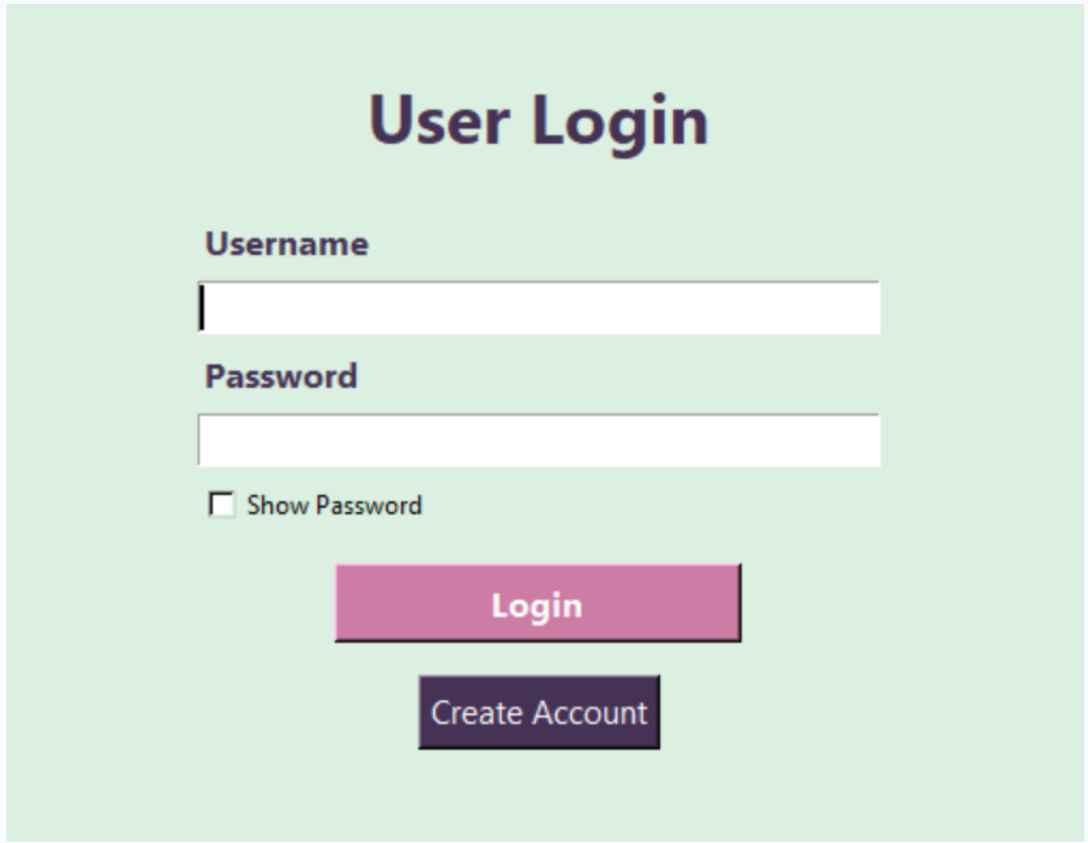
Weak:	**Medium:**	**Strong:**
Less rules satisfied	Some rules satisfied	All rules satisfied

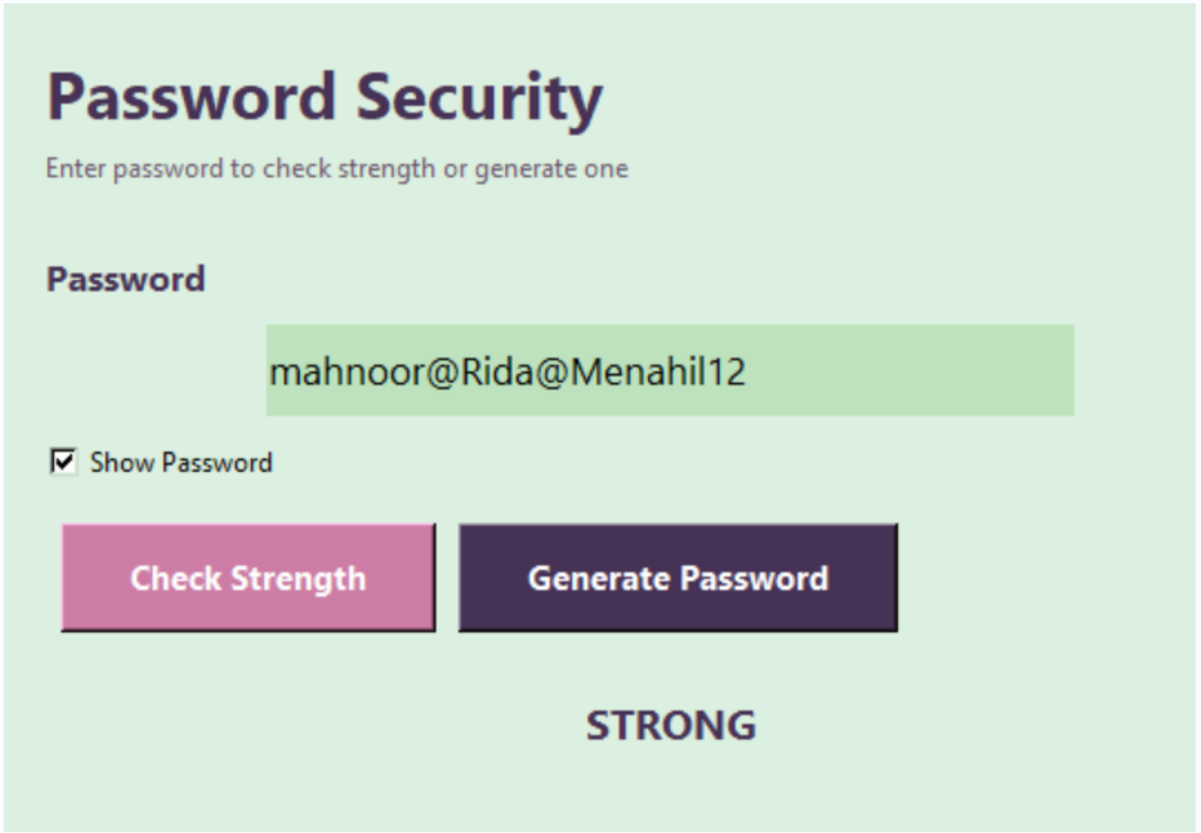
Password Generator

Creates 12-character passwords

Uses letters, numbers, symbols

UI Design:


User Login
Username
Password
Show Password
Login
Create Account


Password Security
Enter password to check strength or generate one
Password
mahnoor@Rida@Menahil12
Show Password
Check Strength
Generate Password
STRONG

Tools & Techniques

- Python
- Tkinter (GUI)
- hashlib (SHA-256)
- re (regex)
- random & string
- CSRF tokens
- Session management

System working

- User signs up
- Password is hashed
- User logs in
- Tokens are generated
- Password is checked or generated
- Actions are logged

Results & limitations

Results:

- Secure login
- Hashed password storage
- Strong password generation

Limitations:

- Uses text files
- Basic hashing method
- Desktop application

Future Improvements:

- Use bcrypt or Argon2
- Encrypt data using AES
- Use database instead of text files
- Add multi-factor authentication

Conclusion:

- Helps create strong passwords
- Improves security
- Educates users

Thank you!!