SecurePass Pro Project

Prepared by: Lawand Rebwar & Mohammed Abubakr Abdulla

Programming Language: Python

1. Introduction

In the modern digital era, weak passwords remain one of the leading causes of security breaches. SecurePass Pro was developed to help users generate strong passwords and analyze their security based on established best practices.

2. Project Objectives

•Generate strong and secure passwords
•Analyze password strength
•Educate users on password security
•Provide a multi-language user interface
•Ensure simplicity and usability

3. System Overview

SecurePass Pro is a standalone desktop application designed to help users create and evaluate secure passwords without storing any sensitive data.

4. Features

• Password generation using letters, numbers, and symbols
• Password strength checking
• Show/Hide password feature
• Copy to clipboard
• Clear input fields
• Multi-language support

5. User Interface Design

The application uses a dark-themed professional interface to reduce eye strain and improve usability. Buttons and input fields are clearly organized for intuitive interaction.

6. Security Considerations

No passwords are stored permanently. The application runs offline and uses random generation methods to ensure user privacy and data security.

7. Technologies Used

- Python Programming Language
- Tkinter / CustomTkinter for GUI
- Python Standard Libraries
- Custom translation module

8. Testing & Validation

The application was tested with weak, medium, and strong passwords to verify correct strength classification and stable performance.

9. Limitations

- No cloud password storage
- No password history saving
- Clipboard does not auto-clear

10. Future Improvements

- Password length customization
- Character selection options
- Strength progress bar
- Password entropy score
- Clipboard auto-clear feature

11. Conclusion

SecurePass Pro is a professional desktop application that demonstrates strong programming skills, security awareness, and user-focused design. It successfully promotes better password practices.

12. References

- NIST Password Guidelines
- OWASP Authentication Best Practices
- Python Official Documentation