Password Strength Checker using C++

A C++ Mini Project

Presented by: Aditya Kadam, Harish Malviya, Saudamini Nayak, Yuvraj

Problem Statement

Our objective is to write a C++ which helps identtify weak passwords and suggest changes to make it strong

Project Objectives

Assess Password Strength

Accurately evaluate passwords based on industry-standard criteria.

Apply OOP Principles

Demonstrate robust software design using Object-Oriented Programming.

Provide Improvement Suggestions

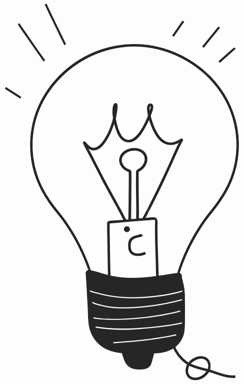
Offer actionable feedback to enhance password security.

Ensure Minimum Security

Help users meet essential security benchmarks for digital accounts.

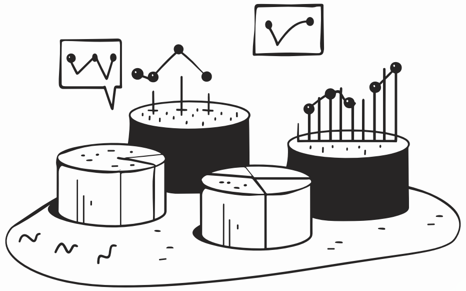
DEVELOPMENT STACK

Technologies Utilized



C++ Language

The core programming language for its performance and system-level capabilities.

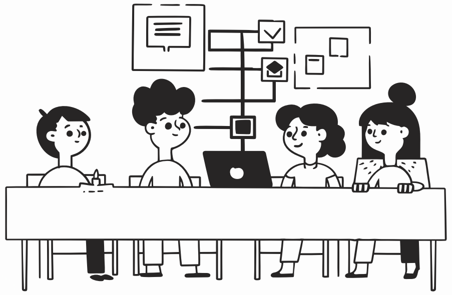


OOP Concepts

Leveraging Classes and Objects for modular, maintainable, and scalable code.

<cctype> Library

Used for character classification functions (e.g., isupper, isdigit, ispunct).



Control Structures

Implementing loops and conditional statements for efficient validation logic.

System Design: Brute-Force Protection

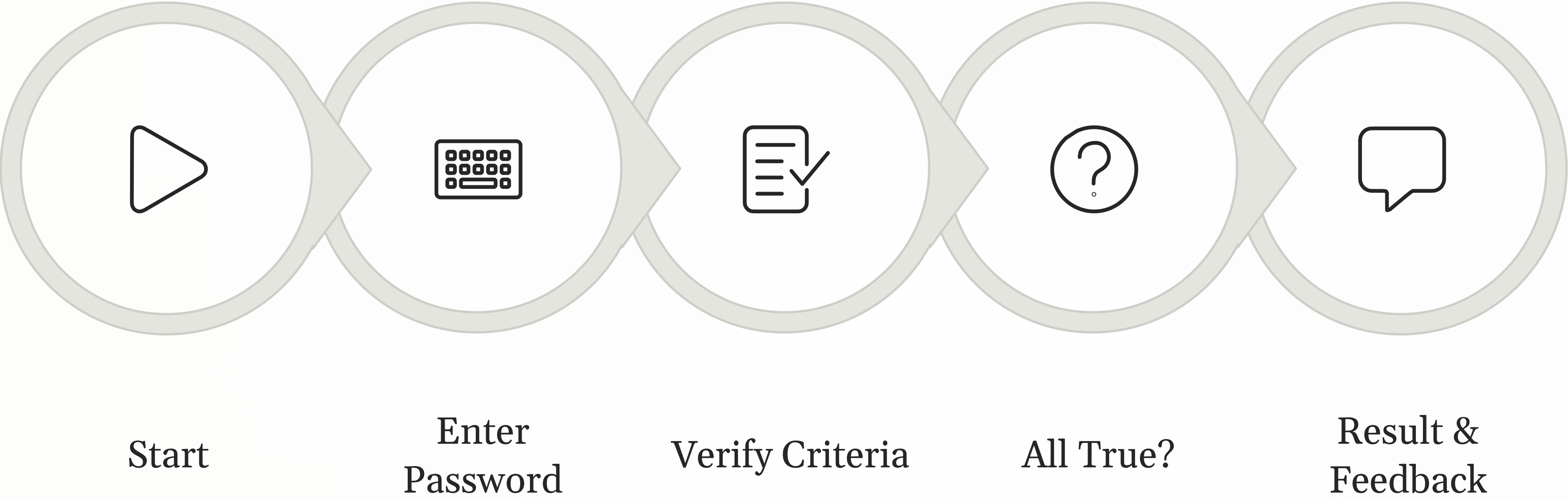
Understanding Brute-Force Attacks

Brute-force attacks systematically try every possible combination of characters until the correct password is found. The time required depends exponentially on password length and complexity.

Password Complexity as Defense

By enforcing diverse character sets (uppercase, lowercase, digits, special characters), we dramatically increase the "keyspace," making brute-force attempts computationally infeasible. This project aims to guide users towards such robust passwords.

Password Checking Flowchart



Start

Enter Password

Verify Criteria

All True?

Result & Feedback

This flowchart illustrates the sequential steps our system follows to evaluate a given password, providing a clear path from input to strength assessment and user feedback.

CODE DEEP DIVE

Code Explanation: Core Logic

The pwcheck Class

Encapsulates all password validation logic, ensuring data integrity and promoting reusability in line with OOP principles.

- Holds password string as private member.
- Methods for validation and suggestion generation.

check() Function

The primary method responsible for orchestrating all validation checks against the provided password.

- Iterates through the password once for efficiency.
- Aggregates individual criteria results.

Validation Logic

Utilizes the <cctype> library to perform granular checks for character types.

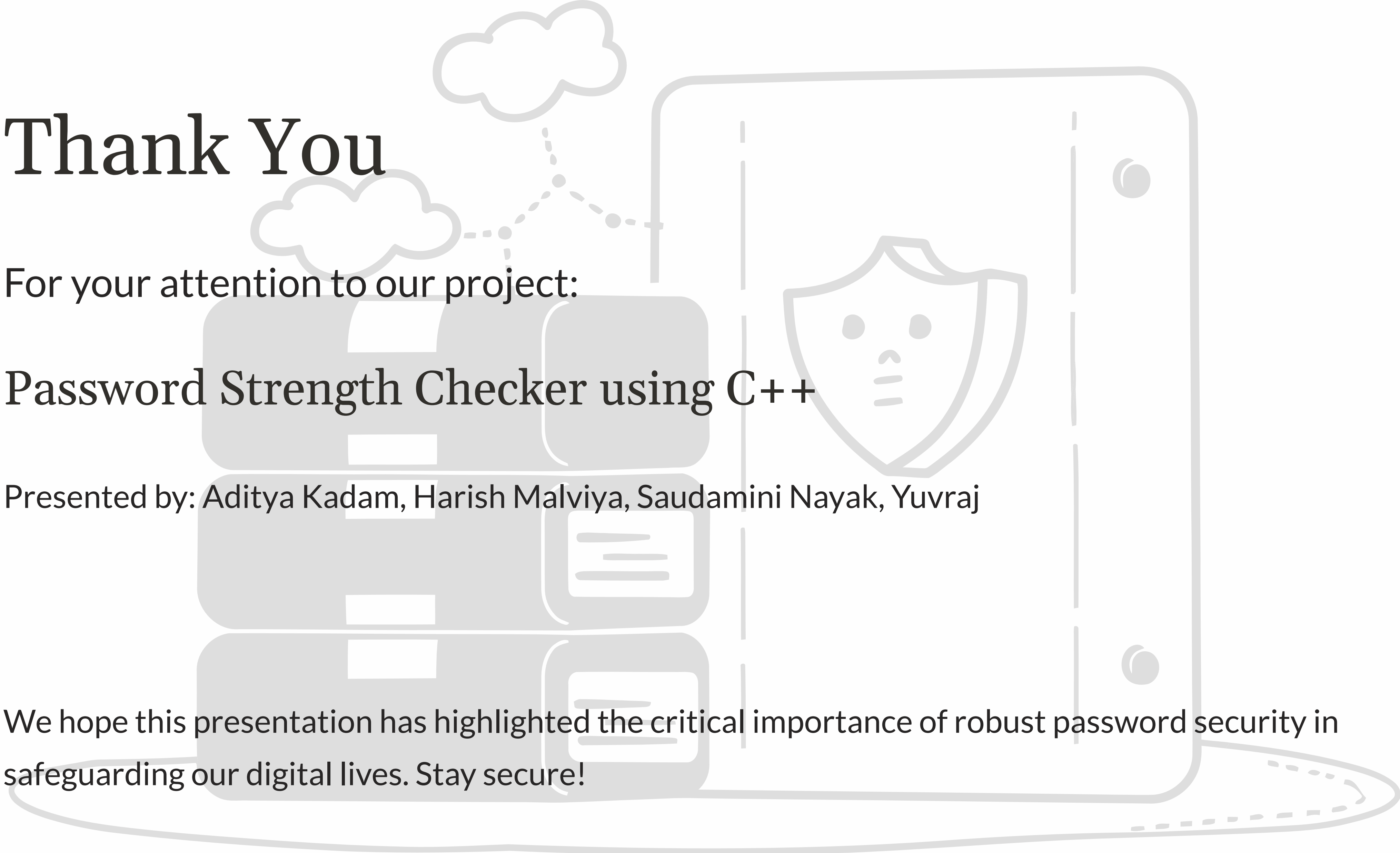
- Minimum length (e.g., 8 characters).
- Presence of uppercase, lowercase, digits, and special characters.

Return Values & Feedback

Communicates the password's strength and guides the user on necessary improvements.

- Returns a strength indicator (e.g., Strong, Medium, Weak).
- Provides specific suggestions for strengthening weak passwords.

Thank You



For your attention to our project:

Password Strength Checker using C++

Presented by: Aditya Kadam, Harish Malviya, Saudamini Nayak, Yuvraj

We hope this presentation has highlighted the critical importance of robust password security in safeguarding our digital lives. Stay secure!