

# Topic: Check Password

My Name

African Institute for Mathematical Sciences, AIMS-Senegal

Supervised by Dr. Yae Olatoundji Ulrich Gaba

October 24, 2025

# Overview

## 1 INTRODUCTION

- Motivation
- Objective

## 2 Adopted Methodology

## 3 Validation Testing

# Overview

## 1 INTRODUCTION

- Motivation
- Objective

## 2 Adopted Methodology

## 3 Validation Testing

**AIMS**African Institute for  
Mathematical Sciences  
SENEGAL

# Introduction

## Motivation

### Context

With the rapid expansion of digital services and online accounts, password security has become a major issue in cybersecurity. Many users still choose weak or predictable passwords, which exposes their accounts to brute-force or dictionary attacks.

**AIMS**African Institute for  
Mathematical Sciences  
SENEGAL

# Introduction

## Motivation

### Context

With the rapid expansion of digital services and online accounts, password security has become a major issue in cybersecurity. Many users still choose weak or predictable passwords, which exposes their accounts to brute-force or dictionary attacks.

### Problem Statement

How can we ensure that a password chosen by a user is secure enough to withstand unauthorized access attempts?



# Overview

## 1 INTRODUCTION

- Motivation
- Objective

## 2 Adopted Methodology

## 3 Validation Testing

**AIMS**African Institute for  
Mathematical Sciences  
SENEGAL

# Introduction

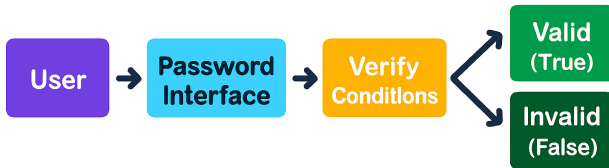
## Objective

To design and develop a Python program capable of analyzing a password entered by a user and automatically determining whether it complies with a predefined set of security rules.

**AIMS**African Institute for  
Mathematical Sciences  
SENEGAL

# Project Workflow

The diagram below illustrates the overall functioning of the password verification program.





# Tools Used

To carry out this project, several Python tools and libraries were employed to ensure both the robustness of the validation logic and the simplicity of the user interface:

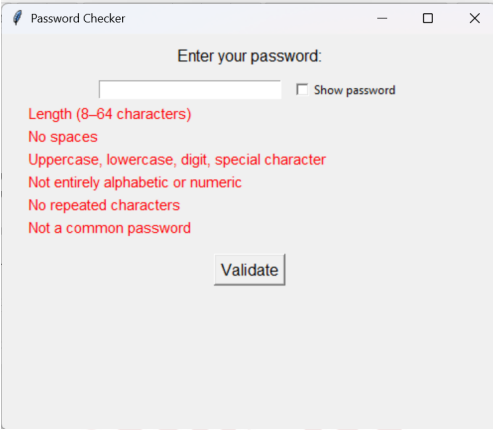
- **Programming Language:** Python 3
- **Library:** Tkinter — Python's standard library for creating intuitive and interactive graphical user interfaces (GUIs)
- **Code Editor:** Jupyter Notebook
- **Execution Environment:** Python Terminal or Tkinter interface

In this section, we will examine how the program behaves depending on the password entered by the user, using code examples.

**Table:** Validation Tests

<b>Password Examples</b>	<b>Results</b>	<b>Error or Confirmation Messages</b>
<b>abc123</b>	False	Missing uppercase letters, special characters, and length $< 8$
<b>qwerty</b>	False	The password is too common and not secure.
<b>AAA111!!!!</b>	False	Contains three identical consecutive characters.
<b>Password1!</b>	True	All security requirements met.





The image shows a window titled "Password Checker" with a standard macOS-style title bar (red, yellow, and green buttons). The window has a light gray background. At the top, it says "Enter your password:". Below this is a text input field. To the right of the input field is a checkbox labeled "Show password". Below the input field, there are five lines of red text representing password requirements: "Length (8–64 characters)", "No spaces", "Uppercase, lowercase, digit, special character", "Not entirely alphabetic or numeric", and "No repeated characters". At the bottom center, there is a button labeled "Validate".

Figure: Example of the user interface — Initial window

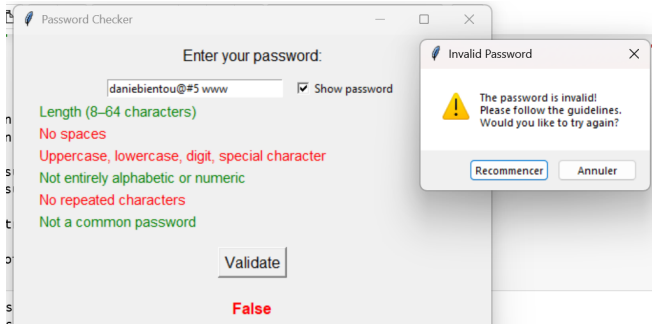


Figure: Example of the user interface — Password analysis

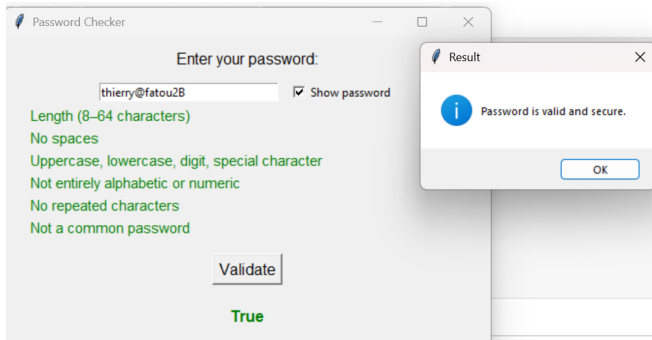


Figure: Example of the user interface — Validation result