Dr. D. Y. PATIL VIDYAPEETH, PUNE
(Deemed to be University)

DR. D. Y. PATIL VIDYAPEETH

PIMPRI, PUNE – 411018

DR. D. Y. PATIL SCHOOL OF SCIENCE & TECHNOLOGY

TATHAWADE, PUNE

A Mini- Project Report of

PASSWORD GENERATOR

SUBMITTED BY :

Name	**:**	**SHAUNAK CHORGE**
Roll No	**:**	**BTAI - 2210**

GUIDED BY :

Mr. Mily Lal

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

ACADEMIC YEAR 2022 – 2023

DR. D. Y. PATIL VIDYAPEETH

PIMPRI, PUNE – 411018

DR. D. Y. PATIL SCHOOL OF SCIENCE & TECHNOLOGY

TATHAWADE, PUNE

CERTIFICATE

This is to certify that the Mini Project

SUBMITTED BY :

Name	**:**	**SHAUNAK CHORGE**
Roll No	**:**	**BTAI – 2210**

Is a bonafide work carried out by the student under the supervision of **Mrs. Mily Lal** and it is submitted towards the partial fulfillment of the requirement Problem Solving by Programming

[illegible]

Mrs Mily Lal
Subject Teacher

[illegible]

Prof. Manisha Bhende
HoD (DYPSST)

ARTIFICIAL INTELLIGENCE & DATA SCIENCE

ACADEMIC YEAR 2022 – 2023

<u>Password Generator</u>

A python mini-project

Problem Statement:

The main problem addressed by the password generator and manager project is the difficulty and inconvenience faced by individuals in creating and managing secure passwords for their numerous online accounts. Weak passwords and password reuse practices put users at risk of unauthorized access and potential data breaches. Remembering multiple complex passwords can also be a burden, leading to the adoption of insecure practices such as using easily guessable passwords or writing them down.

The project aims to tackle these challenges by providing an efficient and secure solution for generating strong passwords and managing them in a centralized manner. It seeks to alleviate the burden of password management, enhance password security, and promote good password hygiene among users

Algorithm

Here's an algorithm outlining the steps for the password generator and manager code:

Step 1: Start the program.

Step 2: Display a welcome message.

Step 3: Prompt the user to enter their name.

Step 4: Prompt the user to enter the desired length of the password.

Step 5: Try the following:

5.1. Convert the input length to an integer.

5.2. Generate a random password of the specified length using a combination of letters, digits, and special characters.

5.3. Display the generated password to the user.

5.4. Save the generated password along with the user's name to a file.

5.5. Display a success message indicating that the password has been saved to the file.

5.6. Retrieve all saved passwords from the file.

5.7. Display a list of saved passwords, including the name and corresponding password.

Step 6: If a ValueError occurs during steps 4 or 5.1:

6.1. Display an error message indicating that the input is invalid.

Step 7: If an OSError occurs while accessing the file in steps 5.4 or 5.6:

7.1. Display an error message indicating that there was an issue with accessing the file.

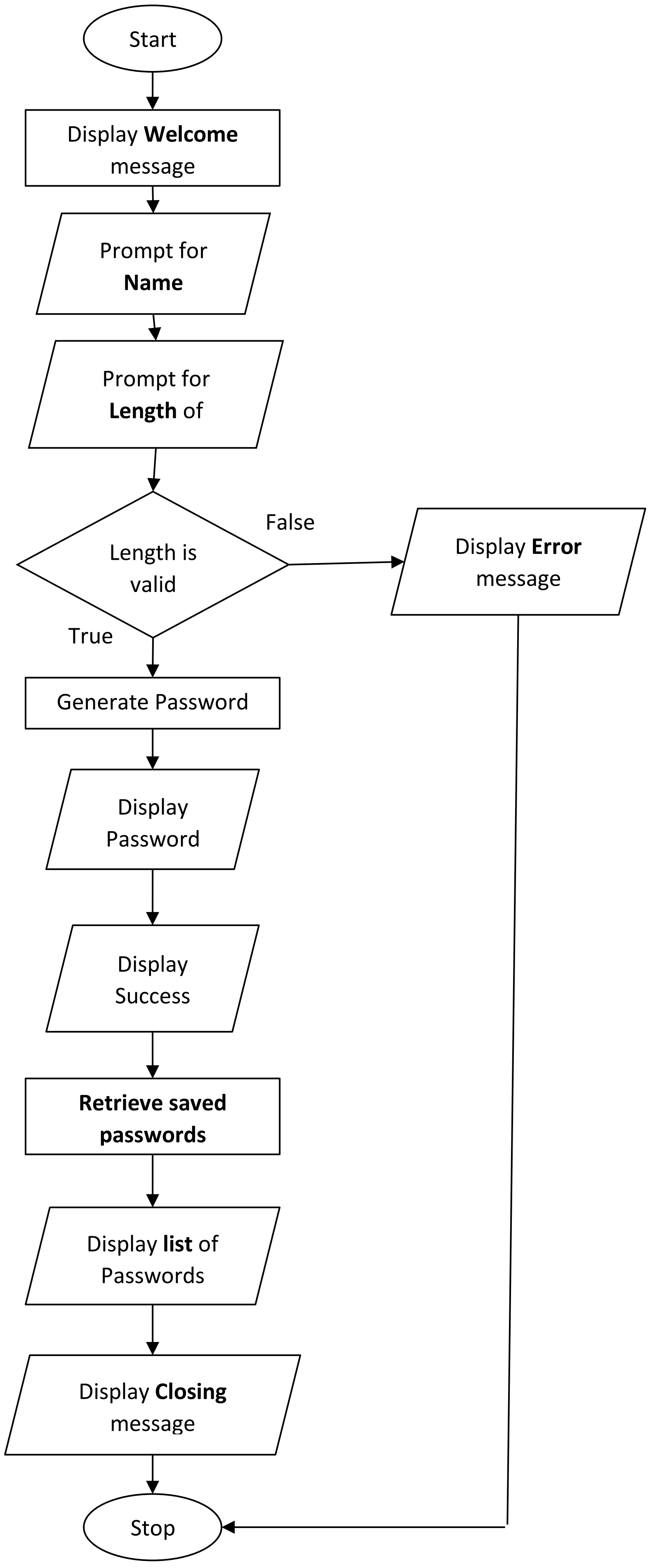
Step 8: If any other exception occurs during steps 4 or 5:

8.1. Display an error message indicating that an unexpected error has occurred.

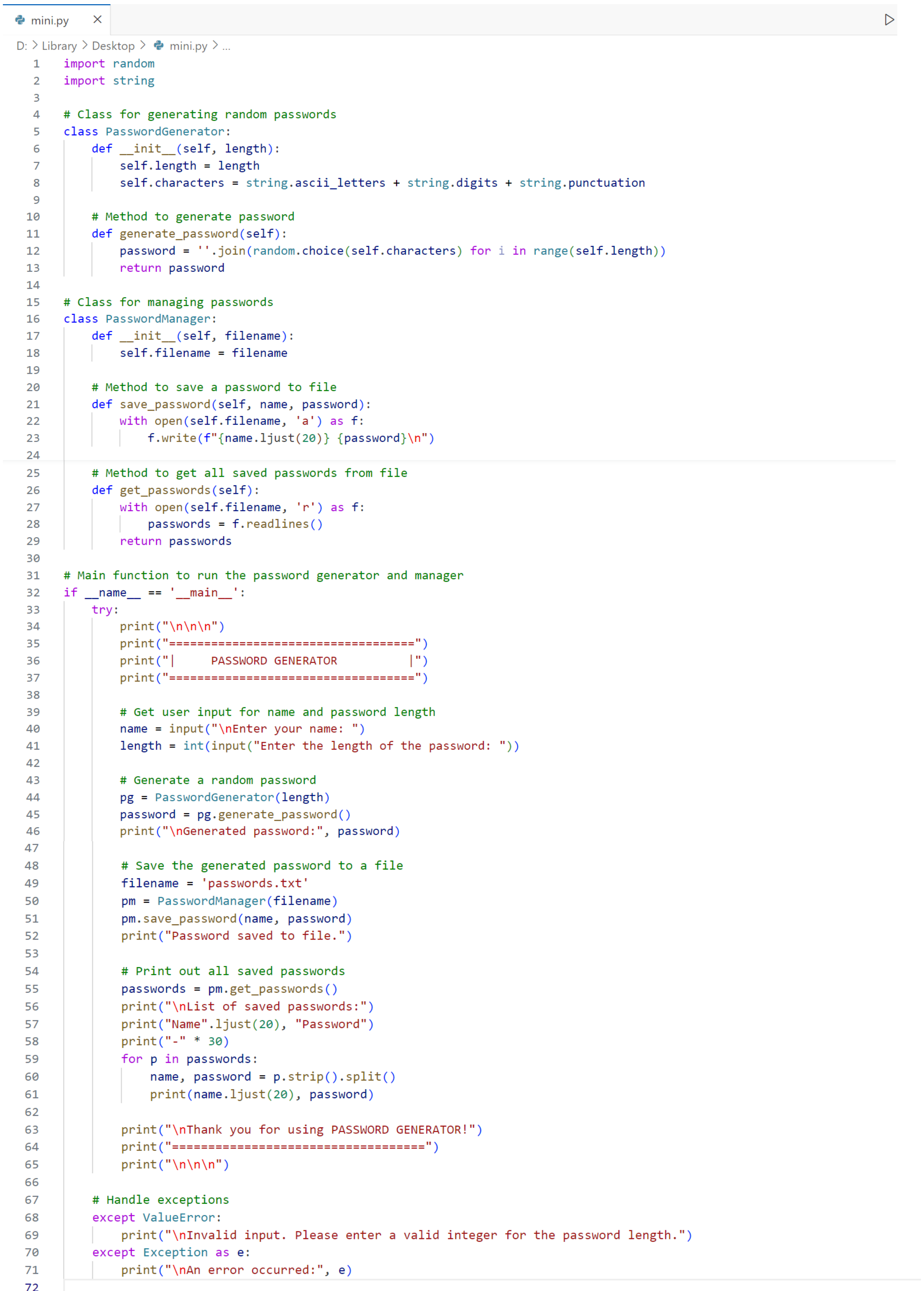
Step 9: Display a closing message.

Step 10: End the program.

Flowchart


Start
Display Welcome message
Prompt for Name
Prompt for Length of
Length is valid
False
True
Display Error message
Generate Password
Display Password
Display Success
Retrieve saved passwords
Display list of Passwords
Display Closing message
Stop

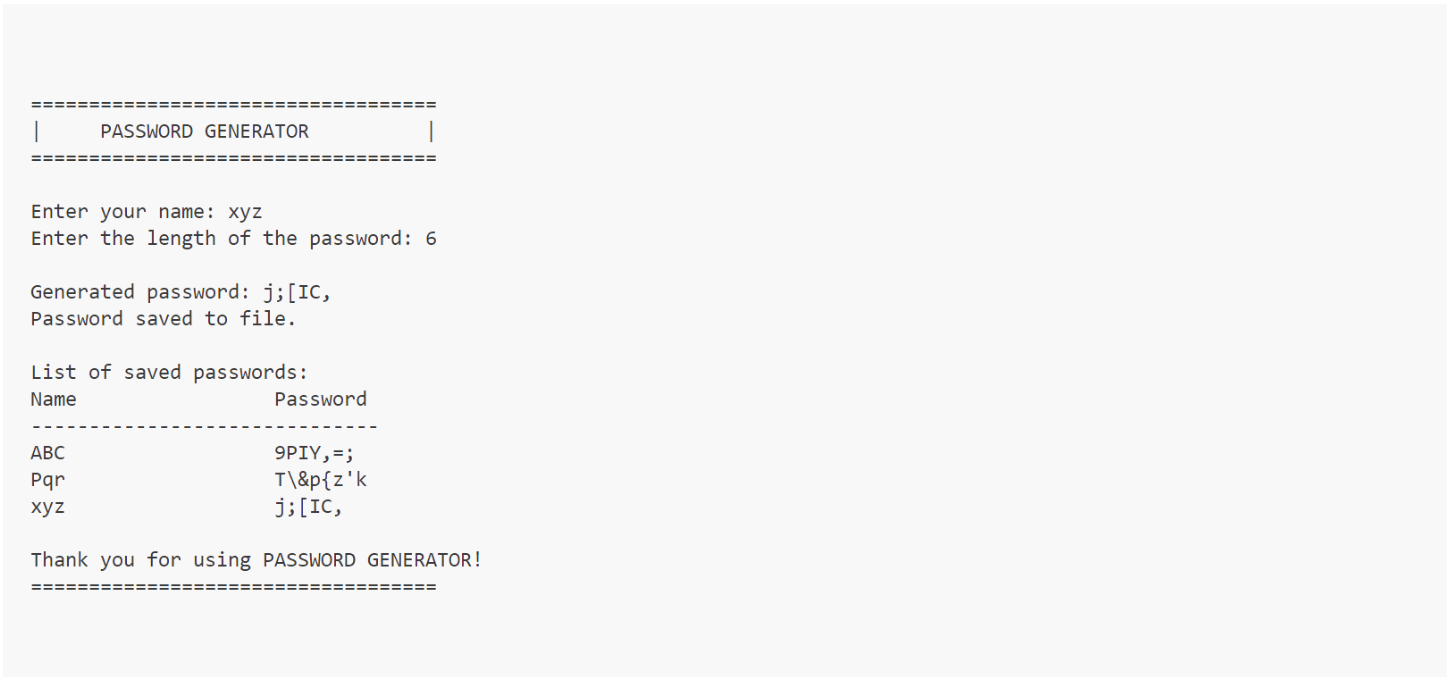
Source Code



```python
import random
import string

# Class for generating random passwords
class PasswordGenerator:
    def __init__(self, length):
        self.length = length
        self.characters = string.ascii_letters + string.digits + string.punctuation

    # Method to generate password
    def generate_password(self):
        password = ''.join(random.choice(self.characters) for i in range(self.length))
        return password

# Class for managing passwords
class PasswordManager:
    def __init__(self, filename):
        self.filename = filename

    # Method to save a password to file
    def save_password(self, name, password):
        with open(self.filename, 'a') as f:
            f.write(f"{name.ljust(20)} {password}\n")

    # Method to get all saved passwords from file
    def get_passwords(self):
        with open(self.filename, 'r') as f:
            passwords = f.readlines()
        return passwords

# Main function to run the password generator and manager
if __name__ == '__main__':
    try:
        print("\n\n\n")
        print("===================================")
        print("|     PASSWORD GENERATOR          |")
        print("===================================")

        # Get user input for name and password length
        name = input("\nEnter your name: ")
        length = int(input("Enter the length of the password: "))

        # Generate a random password
        pg = PasswordGenerator(length)
        password = pg.generate_password()
        print("\nGenerated password:", password)

        # Save the generated password to a file
        filename = 'passwords.txt'
        pm = PasswordManager(filename)
        pm.save_password(name, password)
        print("Password saved to file.")

        # Print out all saved passwords
        passwords = pm.get_passwords()
        print("\nList of saved passwords:")
        print("Name".ljust(20), "Password")
        print("-" * 30)
        for p in passwords:
            name, password = p.strip().split()
            print(name.ljust(20), password)

        print("\nThank you for using PASSWORD GENERATOR!")
        print("===================================")
        print("\n\n\n")

    # Handle exceptions
    except ValueError:
        print("\nInvalid input. Please enter a valid integer for the password length.")
    except Exception as e:
        print("\nAn error occurred:", e)
```

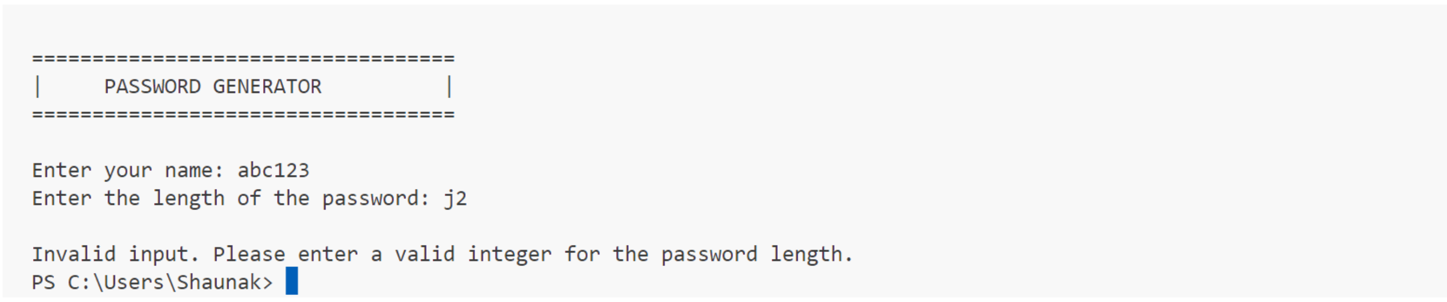
Output

For successful output:



```
===================================
|     PASSWORD GENERATOR          |
===================================

Enter your name: xyz
Enter the length of the password: 6

Generated password: j;[IC,
Password saved to file.

List of saved passwords:
Name                 Password
------------------------------
ABC                  9PIY,=;
Pqr                  T\&p{z'k
xyz                  j;[IC,

Thank you for using PASSWORD GENERATOR!
===================================
```

For expected error:



```
===================================
|     PASSWORD GENERATOR          |
===================================

Enter your name: abc123
Enter the length of the password: j2

Invalid input. Please enter a valid integer for the password length.
PS C:\Users\Shaunak>
```

Conclusion

In conclusion, we are delighted with the password generator and manager project and the valuable contributions it brings to the realm of password security and management. By harnessing the power of Python, we have developed a comprehensive solution that addresses the challenges associated with creating strong passwords and effectively managing them.

By integrating a robust password generator and a secure password storage mechanism, we have empowered users to generate unique and complex passwords for their online accounts. The user-friendly interface and intuitive prompts make the password generation and management processes seamless and hassle-free.

The inclusion of features such as file handling and exception handling has further enhanced the project's functionality and reliability. We are confident that the secure storage of passwords in a designated file ensures the confidentiality and protection of users' sensitive information.

Overall, the password generator and manager project enables users to streamline their password management practices, ensuring the use of strong and unique passwords across all accounts. By prioritizing password security and incorporating essential features into the project, we provide users with a powerful tool to navigate the digital landscape and safeguard their personal information from unauthorized access.

We take pride in the achievements of the password generator and manager project and believe that it will continue to serve as an effective solution for individuals seeking to strengthen their password security and simplify their password management practices.