PASSWORD GENERATOR IN PYTHON

A PROJECT REPORT

Submitted by

AARON MICHEAL RAJ(2303811710421001)

in partial fulfillment for the completion of the course

CGB1121- PYTHON PROGRAMMING

in

COMPUTER SCIENCE AND ENGINEERING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, Affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

MAY, 2024

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report titled **“PASSWORD GENERATOR IN PYTHON”** is the bonafide work of **AARON MICHEAL RAJ (2303811710421001)** who carried out the project under my supervision. Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a course was conferred on an earlier occasion on this or any other candidate.

[illegible]

SIGNATURE

Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

PROFESSOR

Department of CSE

K. Ramakrishnan College of Technology

(Autonomous)

samayapuram-612 112

[illegible]

SIGNATURE

Mrs.S.GAYATHRI,M.E.,

SUPERVISOR

ASSISTANT PROFESSOR

Department of CSE

K. Ramakrishnan College of Technology

(Autonomous)

samayapuram-612 112

Submitted for the viva-voce examination held on 18.06.2024

[illegible]

INTERNAL EXAMINER

DECLARATION

I declare that the project report on **“CONVERTING EMOJI TO TEXT IN PYTHON”** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **“ANNA UNIVERSITY CHENNAI”** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfillment of the requirement of the completion of the course **CGB1121- PYTHON PROGRAMMING**.

Signature

Aaron.

AARON MICHEAL RAJ

Place: Samayapuram

Date:18.06.2024

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project guide **Mrs. S. GAYATHRI, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,**
for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To emerge as a leader among the top institutions in the field of technical education.

MISSION OF THE INSTITUTION

- Produce smart technocrats with empirical knowledge who can surmount the global challenges.
- Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students.
- Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations.

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences

3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations

4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities

   with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

An abstract for a password generator in Python would describe a small software utility that creates random and secure passwords. The generator would typically use Python's random module to produce strings of characters that include uppercase and lowercase letters, digits, and special symbols. The length and complexity of the passwords can usually be customized according to user requirements or security policies. The main goal of such a program is to provide users with strong passwords that are difficult to guess or crack by unauthorized parties.This program includes a main function that interacts with the user to customize the password. It asks for the desired length and whether to include special characters. The generate_password function now has an additional parameter to include or exclude special characters based on user preference. This makes the tool more flexible and user-friendly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

IDE	-	Integrated Development Environment
VS Code	-	Visual Studio Code
re	-	Regular Expression
iOS	-	Iphone Operating System
NLP	-	Natural Language Processing
GUI	-	Graphical User Interface
APIs	-	Application Programming Interface

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION TO PYTHON

1.1.1. Overview

Python is a widely-used, high-level programming language renowned for its readability and simplicity, making it an ideal choice for both novice and seasoned programmers. Created by Guido van Rossum and released in 1991, Python's core philosophy emphasizes code readability and straightforward syntax, allowing developers to write clear and concise code more efficiently compared to other languages like C++ or Java.

1.1.2. Programming Paradigms

Python supports various programming paradigms, including procedural, object-oriented, and functional programming. This flexibility, combined with a dynamic type system and automatic memory management, facilitates the development of a wide range of applications, from simple scripts to complex software systems.

1.1.3. Standard Library

The language's comprehensive standard library, often referred to as "batteries-included," provides built-in modules and functions for handling many programming tasks, such as file I/O, system calls, and even web services. This extensive library helps streamline the development process by offering ready-to-use solutions for common programming challenges.

1.1.4. Third-Party Libraries And Frameworks

One of Python's significant strengths is its extensive ecosystem of third-party libraries and frameworks. Popular libraries such as NumPy and Pandas enable efficient data manipulation and analysis, while frameworks like Django and Flask streamline web development. In the realm of machine learning and artificial intelligence, libraries like TensorFlow and PyTorch are widely adopted for building and deploying sophisticated models.

1.1.5. Versions Of Python

Python has undergone significant evolution since its inception, with two major versions in use today:

Python 2: Released in 2000, Python 2.x series was a major milestone and widely used for many years. However, it reached its end of life on January 1, 2020, and is no longer maintained.

Python 3: Introduced in 2008, Python 3.x series brought substantial improvements and changes to the language, such as better Unicode support, a more consistent syntax, and enhanced standard libraries. Python 3 is the recommended version for all new projects.

1.1.6. Python Tools

Python's ecosystem includes numerous tools that enhance productivity and development experience:

- **IDEs and Code Editors:** Popular options include PyCharm, VS Code, and Jupyter Notebook, which offer features like syntax highlighting, code completion, and debugging.
- **Package Management:** Tools like pip and conda facilitate the installation and management of Python libraries and dependencies.
- **Virtual Environments**: virtualenv and venv allow developers to create isolated environments for different projects, ensuring dependency conflicts are avoided.
- **Testing Frameworks:** unittest, pytest, and nose are commonly used for writing and running tests to ensure code reliability and correctness.
- **Build Tools:** setuptools and wheel help in packaging Python projects, making them easy to distribute and install.
- **Documentation Generators:** Tools like Sphinx are used to create comprehensive documentation for Python projects.
- **Linters and Formatters:** pylint, flake8, and black help maintain code quality and consistency by enforcing coding standards and formatting.

1.1.7. Versatility And Adoption

Python's simplicity and versatility have led to its widespread adoption in various fields, including web development, data science, artificial intelligence, automation, and scientific computing. Its active community continually contributes to a rich repository of resources, tutorials, and documentation, making it easier for developers to learn and apply Python effectively.

CHAPTER 2
PROJECT DESCRIPTION

2.1. PROJECT INTRODUCTION

The password generator in Python is a software tool that automates the creation of secure and complex passwords. It leverages Python's capabilities to generate random sequences of characters, including uppercase and lowercase letters, numbers, and special symbols. This tool is essential in an era where digital security is paramount, as it helps users generate passwords that are difficult for attackers to guess or brute-force. With its user-friendly design and customization options, the password generator is an invaluable asset for anyone looking to bolster their online security.

2.2. PROJECT OBJECTIVE

The password generator in Python is designed with the objective of bolstering digital security by providing a reliable means to create strong, unique passwords. Its primary goal is to safeguard users against common cyber threats, such as dictionary attacks and brute force, by automating the generation of complex passwords. This tool simplifies the process of obtaining new passwords, thereby discouraging the reuse of passwords across different accounts and enhancing overall password management practices. With customization options available, it caters to a variety of security needs and user preferences, ultimately promoting safer online experiences through improved password security.

2.3. PROBLEM STATEMENT

The problem statement for a password generator in Python addresses the issue of weak password creation and management, which is a significant vulnerability in cybersecurity. Despite widespread knowledge of the risks associated with simple and commonly used passwords, many users continue to employ inadequate passwords due to the difficulty of remembering complex ones. The password generator aims to resolve this by automatically creating strong, random passwords that are difficult to guess or hack, thus significantly reducing the risk of unauthorized access to user accounts and sensitive information.

2.4 LIBRARIES USED

In the password generator program provided earlier, the following Python libraries are used:

- **string:** This library provides a set of characters including letters, digits, and punctuation which can be used to create passwords.
- **random:** This library is used to generate random selections, ensuring that the passwords produced are unpredictable and unique.

These standard libraries come built-in with Python, so there’s no need for additional installation to use them in the password generator program.

CHAPTER 3
SYSTEM ANALYSIS

3.1. EXISTING SYSTEM

The existing system of the password generator program in Python, as described earlier, is a command-line application that generates a random password based on user input. It utilizes Python's built-in libraries to create a sequence of characters that includes uppercase and lowercase letters, numbers, and optionally, special symbols. The user can specify the length of the password and whether to include special characters. The current system is functional and provides a basic level of customization, allowing users to generate passwords that meet their security requirements.

3.1.1. DISADVANTAGES

The existing password generator system in Python, while functional, has several disadvantages:

1. It operates on the command-line, which may not be user-friendly for those unfamiliar with terminal interfaces.
2. There is no built-in functionality to store or manage the generated passwords securely.
3. The program does not provide a password strength rating, leaving users to judge the security of their passwords themselves.
4. It lacks integration with web browsers or other applications for autofill purposes.
5. The program does not offer a graphical user interface (GUI), which could limit its accessibility to a broader audience.

These limitations suggest areas where the password generator could be improved to enhance user experience and security.

3.2. PROPOSED SYSTEM

The proposed upgrade for the password generator project envisions a more sophisticated system that enhances both usability and security. It would feature a graphical user interface (GUI) to facilitate ease of use for all users, regardless of their technical expertise. Secure storage mechanisms, potentially incorporating encryption, would be introduced to manage the generated passwords safely. A built-in password strength meter would provide immediate feedback on the security level of passwords, guiding users towards stronger options. Integration with web browsers and other applications would enable autofill functionality, streamlining the login process. Additionally, cloud synchronization capabilities would ensure that users can access their passwords from any device, providing convenience and flexibility. This proposed system aims to transform the password generator into a comprehensive security tool that caters to modern digital needs.

3.2.1. ADVANTAGES

The proposed system for the password generator project offers several advantages:

1. The graphical user interface (GUI) will make it more accessible and easier to use for a broader audience, including those less familiar with command-line tools.
2. Secure password storage with encryption will provide users with a safe way to keep track of their passwords.
3. The password strength meter will help users create stronger passwords, thereby enhancing their overall cybersecurity.
4. Browser and application integration for autofill will streamline the login process, saving time and improving user experience.
5. Cloud synchronization will allow users to access their passwords from any device, ensuring convenience and continuity across platforms.

These advantages position the proposed system as a comprehensive solution for secure password generation and management in today’s digital landscape.

CHAPTER 4
SYSTEM DESIGN & MODULES

4.1. BLOCK DIAGRAM

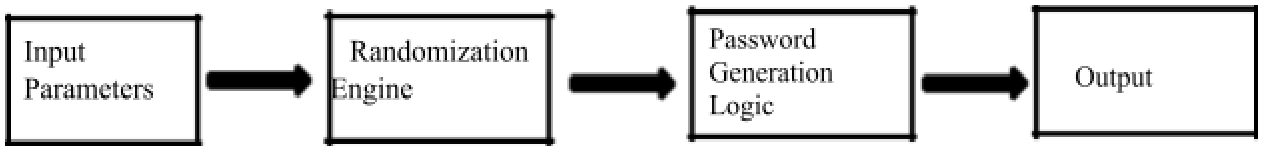


Fig. 4.1. Block Diagram

4.2. MODULE DESCRIPTION

In the proposed system for the password generator project, the following modules could be described:

4.2.1 GUI Module: This module would handle the graphical user interface, providing a visual platform for users to interact with the password generator.

4.2.2 Password Generation Module: This core module would use Python's random library to generate secure passwords based on user-defined criteria such as length and character types.

4.2.3 Password Strength Module: This module would assess the strength of generated passwords using various metrics and provide users with a visual strength indicator.

4.2.4 Storage Module: Responsible for securely storing and managing passwords, this module might implement encryption to protect sensitive data.

4.2.5 Integration Module: This module would facilitate the integration of the password generator with web browsers and applications for autofill functionality.

4.2.6 Synchronization Module: This module would manage cloud synchronization, allowing users to access their passwords across different devices.

Each module plays a crucial role in enhancing the functionality and security of the password generator system.

CHAPTER 5
CONCLUSION & FUTURE ENHANCEMENT

5.1. CONCLUSION

The password generator in Python serves as a robust tool for creating secure and unique passwords, crucial for maintaining digital security. By leveraging Python's randomization capabilities and character sets, it generates passwords that are difficult to predict or breach. This project underscores the importance of strong password policies in safeguarding user data across various platforms.

The Python-based password generator is a vital tool for cybersecurity, providing users with strong, random passwords. It mitigates risks of unauthorized access, enhancing protection for digital identities and sensitive data. The simplicity of Python makes the tool adaptable and efficient. As digital threats evolve, such tools become indispensable in personal and professional contexts, promoting better security habits among users.

5.2. FUTURE ENHANCEMENT

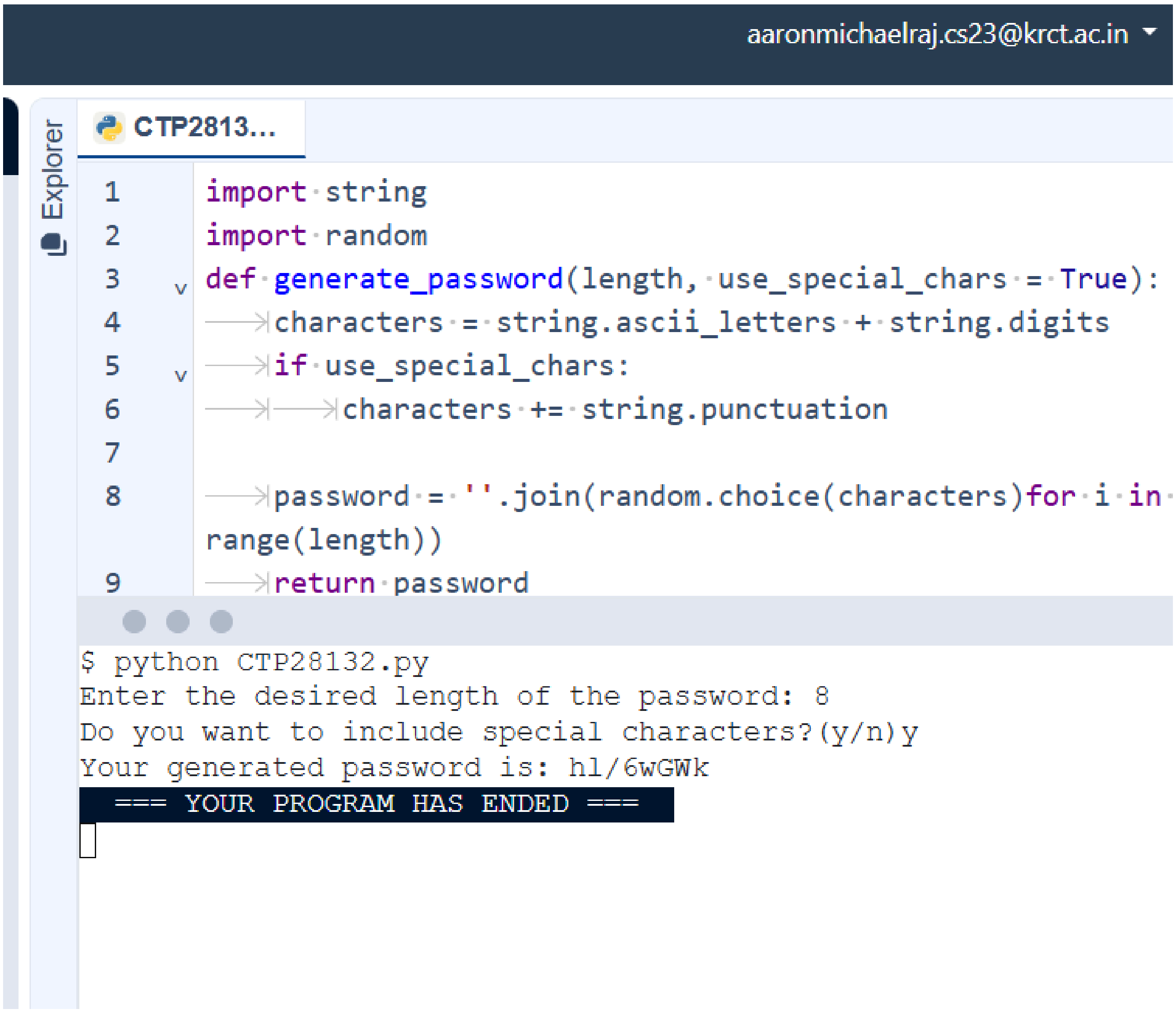
The Python password generator project can be expanded with several enhancements to increase its utility and security. Future updates could introduce a graphical interface for ease of use, cloud synchronization for managing passwords across various devices, and multilingual support to serve a wider audience. Additional features might include password strength indicators to encourage stronger passwords, customizable character sets for tailored security, and a mobile app for accessibility on the go. To further secure password retrieval, two-factor authentication could be implemented. Browser extension integration would streamline the autofill process, and options for pronounceable or pattern-based passwords would add versatility. Continuous refinement based on user feedback and evolving security standards will ensure the password generator remains a reliable tool in the cybersecurity arsenal.

APPENDICES

APPENDIX A-SOURCE CODE

```
 import string

 import random



 def generate_password(length,
use_special_chars=True):

     # Define the characters that will be used to generate
the password

     characters = string.ascii_letters + string.digits

     if use_special_chars:

         characters += string.punctuation



     # Create a random password

     password = ''.join(random.choice(characters) for i in
range(length))



     return password



 def main():

     # User input for customization

     length = int(input("Enter the desired length of the
password: "))

     use_special_chars = input("Do you want to include
special characters? (y/n): ").lower() == 'y'


    # Generate and print the password

    print("Your generated password is:",
generate_password(length, use_special_chars))


 if __name__ == "__main__":

    main()
```

APPENDIX B -SCREEN SHOTS



```
import string
import random
def generate_password(length, use_special_chars = True):
	characters = string.ascii_letters + string.digits
	if use_special_chars:
		characters += string.punctuation

	password = ''.join(random.choice(characters)for i in range(length))
	return password
```

```
$ python CTP28132.py
Enter the desired length of the password: 8
Do you want to include special characters?(y/n)y
Your generated password is: hl/6wGWk
 === YOUR PROGRAM HAS ENDED ===
```