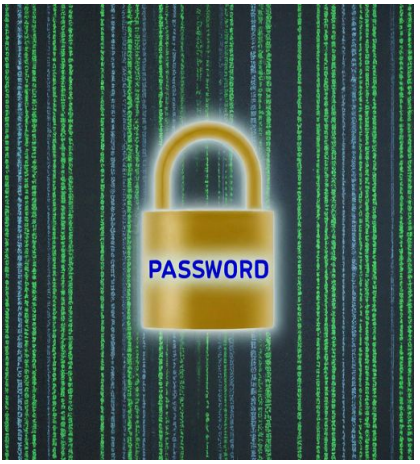
Mini-Project: Password Generator



Done by: Pramodh Narain
Sem: 2nd Semester
Department:
CSE (Data Science)
Batch: 2023-2027

TABLE OF CONTENTS

INTRODUCTION

In the digital age, the importance of strong and secure passwords cannot be overstated. Passwords serve as the first line of defense against unauthorized access to personal and sensitive information. The Password Generator aims to address this critical need by providing a tool that can generate complex and unique passwords based on user-defined criteria/ This project utilizes Python to create a customizable password generator that allows users to specify the number of letters, symbols, and numbers in their passwords, By ensuring the randomness and complexity of the generated passwords, this project helps enhance security for various applications, from online accounts to confidential documents. The flexibility and ease of use the tool make it a valuable resource for anyone looking to improve their password security practices.

SYSTEM REQUIREMENTS

To run the Password Generator Project, the following system requirements must be met:

1. **Operating System:** Windows, macOS, or Linux.
2. **Python Version:** Python 3.6 or higher.
3. **Memory:** At least 512 MB of RAM.
4. **Storage:** At least 10 MB of free disk space.

CODING TOOLS:

The following coding tools and libraries are used in this project:

1. **Python:** The primary programming language
2. **Random Module:** Part of the Python Standard Library, used for generating random selections and shuffling.

MAJOR SECTIONS/FEATURES:

Following are the major sections of our Python Password Generator Project:

1. **User Input:**

   The user is prompted to enter the desired number of letters, symbols, and numbers for the password.

2. **Password Generation:**

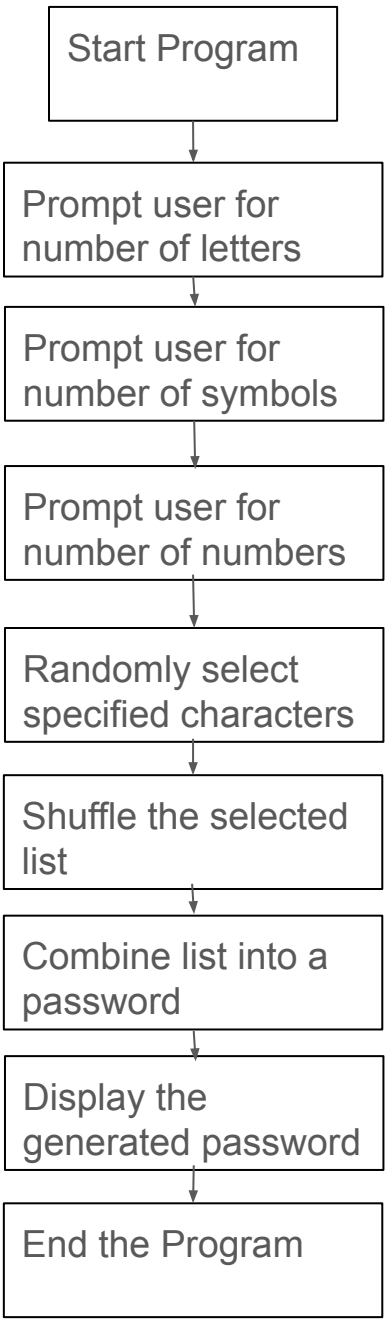
- Random selection of characters from predefined lists of letters, numbers, and symbols.
- Combination of these characters into a list.

3. **Shuffling and Outpuut:**

- Shuffling the list of selected characters to ensure randomness.
- Combining the shuffled list into a string and displaying the final password.

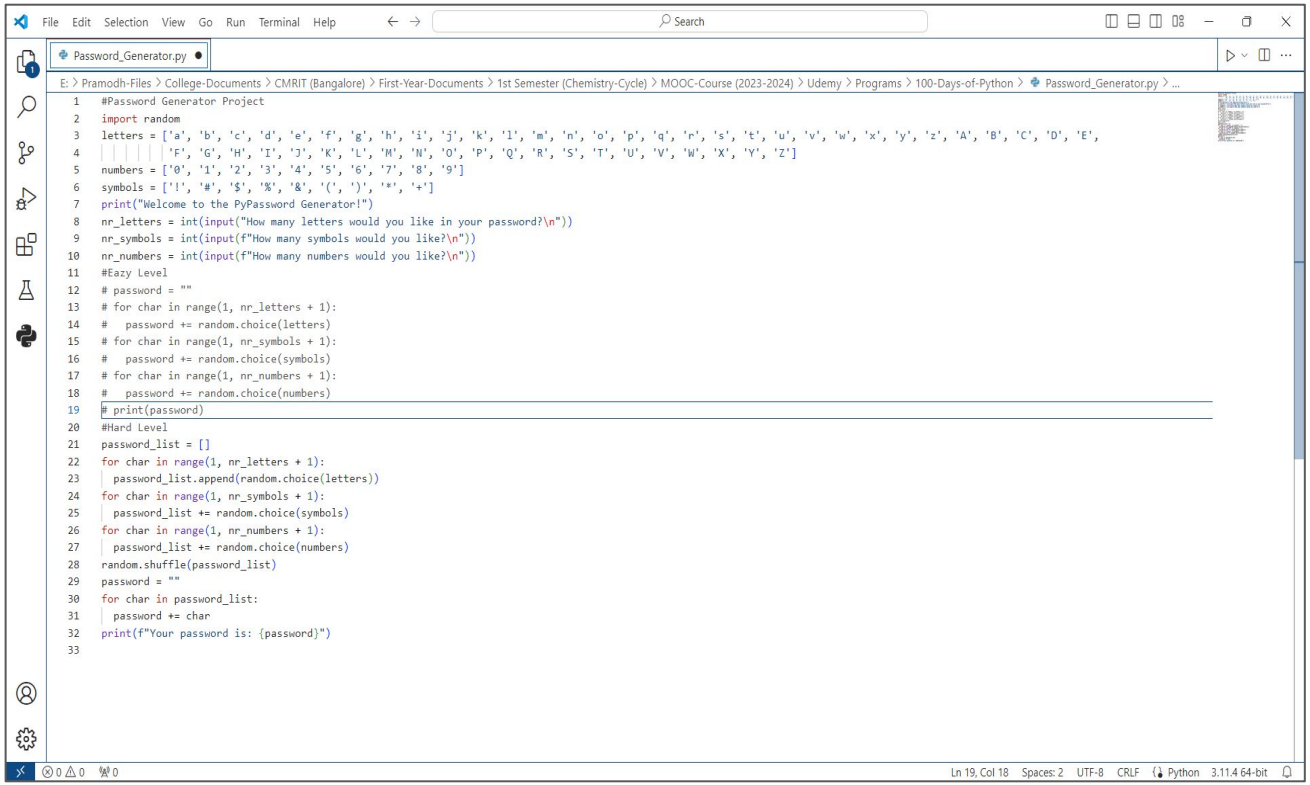
PROJECT MAP

The following flowchart illustrates the workflow of the Password Generator Project:



SOURCE CODE:

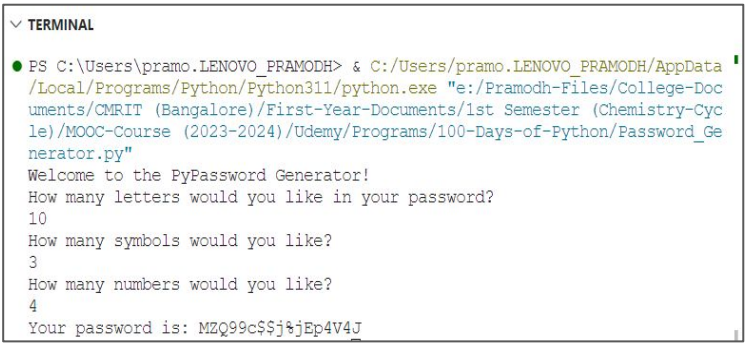
The source code of our Python Password Generated Project is shown as follows:



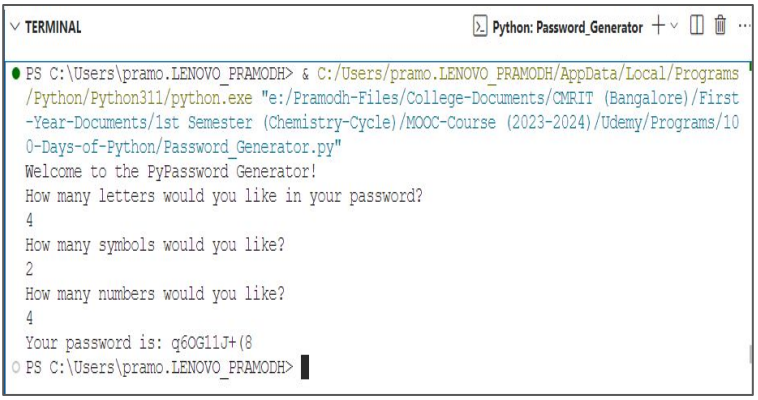
OUTPUT AND SCREENSHOTS

Following are some sample outputs generated by our Python Password Generator Program:

1. Sample-Output-1:



2. Sample Output-2:



CONCLUSION:

The Password Generator Project successfully demonstrates the creation of a customizable and secure password generator using Python. By allowing users to define the composition of their passwords, the project provides flexibility and enhances security. The user of the ‘random’ module ensures that the generated passwords are both random and strong, suitable for protecting personal and sensitive information.

REFERENCES USED FOR THIS PROJECT:

1. **Python Standard Library Dosumentation:**

   ‘Random’ Module:

   https://docs.python.org/3/library/random.html

2. **Python.org:**

   https://www.python.org/

3. **General Information on Password Security:**

   https://www.csoonline.com/article/3244707/what-is-password-security-and-how-to-manage-passwords-safely.html