
# INT213 PROJECT REPORT

Report

Submitted in Partial Fulfilment of the

Requirement for Award of the Degree

Of

**B.TECH COMPUTER SCIENCE ENGINEERING**

By

Udayan Debnath (12105945)

Roll No:-A85

Sec:-K20MH

Under the Guidance of

**Dr SUKHVIR KAUR**

Department of CSE

**LOVELY PROFESSIONAL UNIVERSITY**

**PHAGWARA, PUNJAB(INDIA)-144411**

**2021-22**

# CERTIFICATE

Certified that mini project work entitled "**PASSWORD GENERATOR**" is a Bonafede work carried out in the 3rd semester by "**UdayanDebnath"** in fulfilment for the award of B.TECH in Computer Science and Engineering from School of CSE, Lovely Professional University, Jalandhar, Punjab during the academic year 2021-2022.

# ACKNOWLEDGEMENT

Foremost, we would like to express our sincere gratitude to our advisors Dr SUKHVIR KAUR for the continuous support in our studies and project. Their guidance helped us all the time to complete this minor project. We could not have imagined having better project mentors for our "**PASSWORD GENERATOR**" project. We would also like to thank our parents, brothers and sisters for having given us their undisputable support throughout, as always, for which our expression of thanks likewise does not suffice. Also, we thank our friends and classmates for supporting us.

# INDEX

# INTRODUCTION

In a recent report, it's shown that over 80% of breaches related to hacking are a result of stolen or weak passwords. Creating strong and secure passwords is an important step to protect our personal information. Let's start developing this amazing project that helps in generating random passwords and also learn some concepts of python. So here is the Python password Generator project.

# OBJECTIVES

The objective of this project is to generate passwords in Python. It requires three modules: Tkinter , random and string. Knowledge of functions in python is recommended in this project.

This is a very simple project in which we just have to enter the length of the password and select the password strength from the given options. After entering these two things a password will be generated randomly according to our requirement at the bottom of the screen.

# FEATURES OF A PASSWORD GENERATOR

1. Adjust guidelines to fit different sites' unique password requirements
2. Generate strong passwords using secure technology with built-in randomness
3. Are integrated into a password manager like Dashlane to create, manage, and easily use all your strong passwords

# Project File Structure

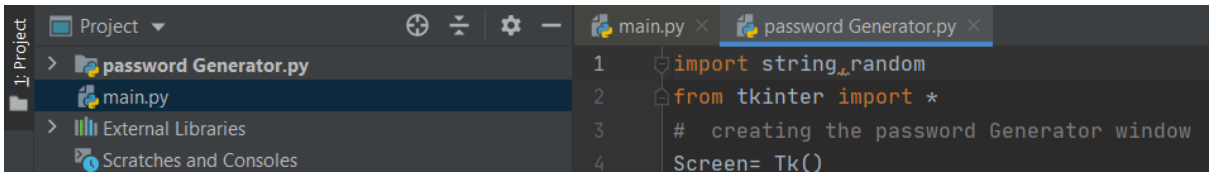
Steps to follow to create a password Generator in Python:

1. Importing Libraries
2. Initializing Window
3. Program Title
4. Create Radio Buttons
5. Length of Password
6. Logic of the Program
7. Creating a button to Generate Password

Let's start the password generator project in Python:

## 1. Importing Libraries

To start with the project the first step is to import libraries.




**Code Explanation:**

a. **tkinter =** It is the most commonly used technique to create Graphical User Interface.
**b. random =** This module generates random numbers.
**c. string =** This module helps to manipulate strings with the help of additional tools.

## 2. Initializing project window

```
Scratches and Consoles
```

```
4   Screen= Tk()
5   Screen.geometry("500x300")
6   Screen.title("password Generator by Udayan")
7   #program title
8   Title = StringVar()
```

**Code Explanation:**
**a. Tk():** It's the standard interface in python.
**b. geometry():** Size, Position, and other attributes of the layout of the screen are decided by this function.
**c. title():** The first letter of every word in the string is converted to capital letter with the help of this function.

## 3. Program Title

```
7   #program title
8   Title = StringVar()
9   TitleLabel = Label(Screen, textvariable=Title).pack()
10  Title.set("Choose Strength of Password-")
```

**Code Explanation:**
**a. StringVar():** It holds a string.
**b. Label():** It is used to display text, images and even bitmaps.
**c. set():** Values of the variable can be set using this method.

## 4. Create Radio Buttons

```
def SelectionOptions():
    SelectionOptions = Choice.get()

Choice= IntVar()
RadioButton1 = Radiobutton(Screen, text="POOR", variable=Choice, value=1, command=SelectionOptions).pack(anchor=CENTER)
RadioButton2 = Radiobutton(Screen, text="AVERAGE", variable=Choice, value=2, command=SelectionOptions).pack(anchor=CENTER)
RadioButton3 = Radiobutton(Screen, text="STRONG", variable=Choice, value=3, command=SelectionOptions).pack(anchor=CENTER)

LabelChoice = Label(Screen)
LabelChoice.pack()

LengthLabel = StringVar()
LengthLabel.set("Password Length")
LengthTitle = Label(Screen, textvariable=LengthLabel).pack()
```

Choice variable contains an integer value. SelectionOptions function made to create radio buttons.

**a. IntVar():** It holds a variable.
**b. Radiobutton:** This widget provides multiple choice to the user.
**c. pack():** The position of widgets is declared by this method in Relation to one another.
**d. anchor:** It is used to define the relative position of text

## 5. Length of Password:

```
#Length of password
Value = IntVar()
SpinLength = Spinbox(Screen, from_=6, to_=25, textvariable=Value, width=14).pack()
```

**Code Explanation:**
**a. Spinbox():** This widget is used to select the length of the password.
From 9 to 25 denotes the length of the password that can be set.

**b. textvariable =** Current text is retrieved to the entry widget.

## 6. Logic Of The Program:

```
#logic of program

Poor = string.ascii_uppercase+string.ascii_lowercase
Average = string.ascii_uppercase+string.ascii_lowercase+string.digits
Symbols="""~`! @#$%^&*()_-+={[}]|\:;"'<,>.?/ """
Advance= Poor+Average+Symbols

def PassGen():
    if Choice.get()==1:
        return "".join(random.sample(Poor,Value.get()))
    if Choice.get()==2:
        return "".join(random.sample(Average,Value.get()))
    if Choice.get()==3:
        return "".join(random.sample(Advance,Value.get()))
```

**Code Explanation:**
Poor variable is set such that the password generated contains a password that is made up of lower and uppercase letters. Average variable generates a password containing lowercase, uppercase letters, and digits. The Advanced variable generates a password that contains symbols, lowercase, uppercase letters and digits.

**a. string.ascii_uppercase:** It consists of all the uppercase letters.
**b. string.ascii_lowercase:** It consists of all the lowercase letters.

**c. string.digits:** It consists of all the digits.
**d. get():** It helps in getting the text written inside the text widget.
**e. join():** It is used to concatenate the string.
**f. random.sample():** It is used to generate random passwords.

# 7. Creating a button to generate password:

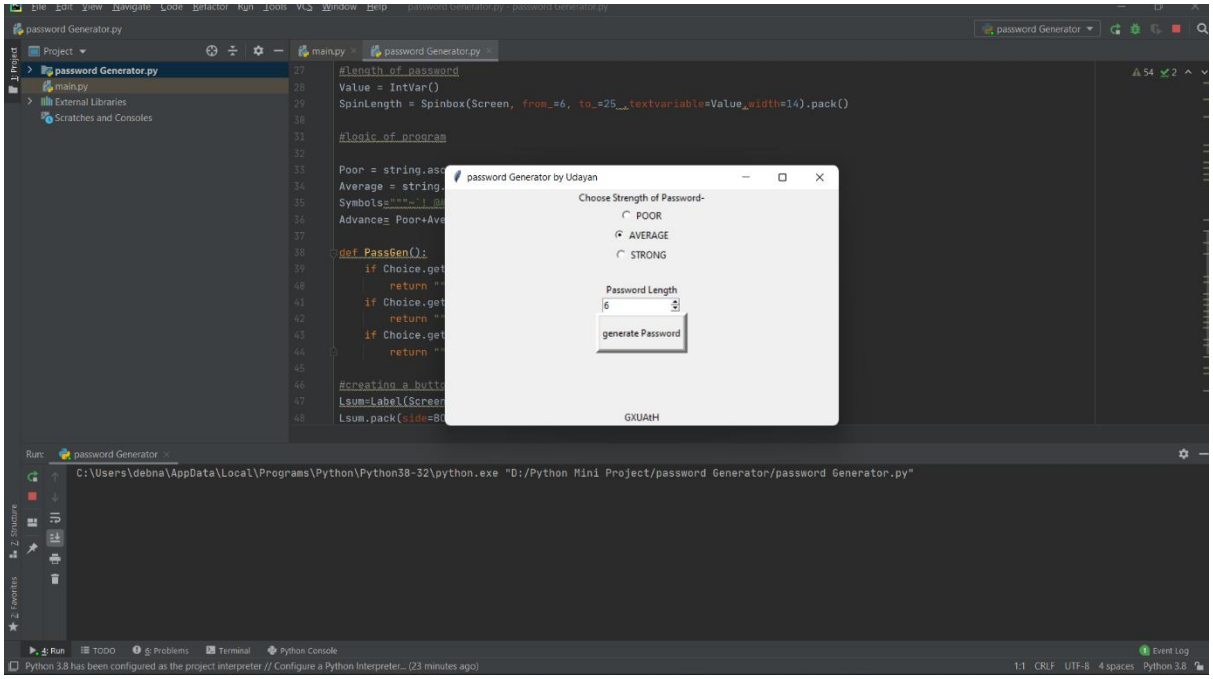
```
#creating a button that will generate password
Lsum=Label(Screen,text="")
Lsum.pack(side=BOTTOM)
def CallBack():
    Lsum.config(text=PassGen())
Password=str(CallBack())
PassGenButton = Button(Screen,text="generate Password",bd=5, height=2,,command=CallBack,pady=3)
PassGenButton.pack()

Screen.mainloop()
```

**Code Explanation:**
PasswordGen variable generates a button. Lsum variable takes the password generated from the function CallBack() and displays it at the bottom of the screen.

**a. config():** It is used to change the property of a widget.
**b. bd:** The size of the border is represented with the help of this option.
**c. pady:** It represents the number of pixels that are needed to pad widget.
**d. mainloop():** It runs the event loop in tkinter.

# Python Password Generator Output:-



# Summary

We have completed the password Generator python project successfully. We used three python modules: tkinter, string, and random, they are required to start developing the project. Few functions are also created to develop this python project