

# PASSWORD GENERATOR PROJECT

DONE BY : ALSAMATH J

```
import random
```

```
def generate_password(length)
```

```
characters =
```

```
"abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01234567  
89!@#$%^&*()_+-=[]{}|;:,.<>?`~"
```

```
# Initialize an empty string to store the password
```

```
password = ""
```

```
# Generate password using a simple for loop
```

```
for _ in range(length):
```

```
    password += random.choice(characters)
```

```
return password
```

```
# Ask user for password length
```

```
while True:
```

```
    try:
```

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

```
        if length <= 0:
```

```
            print("Please enter a positive integer.")
```

*else:*

*break*

*except ValueError:*

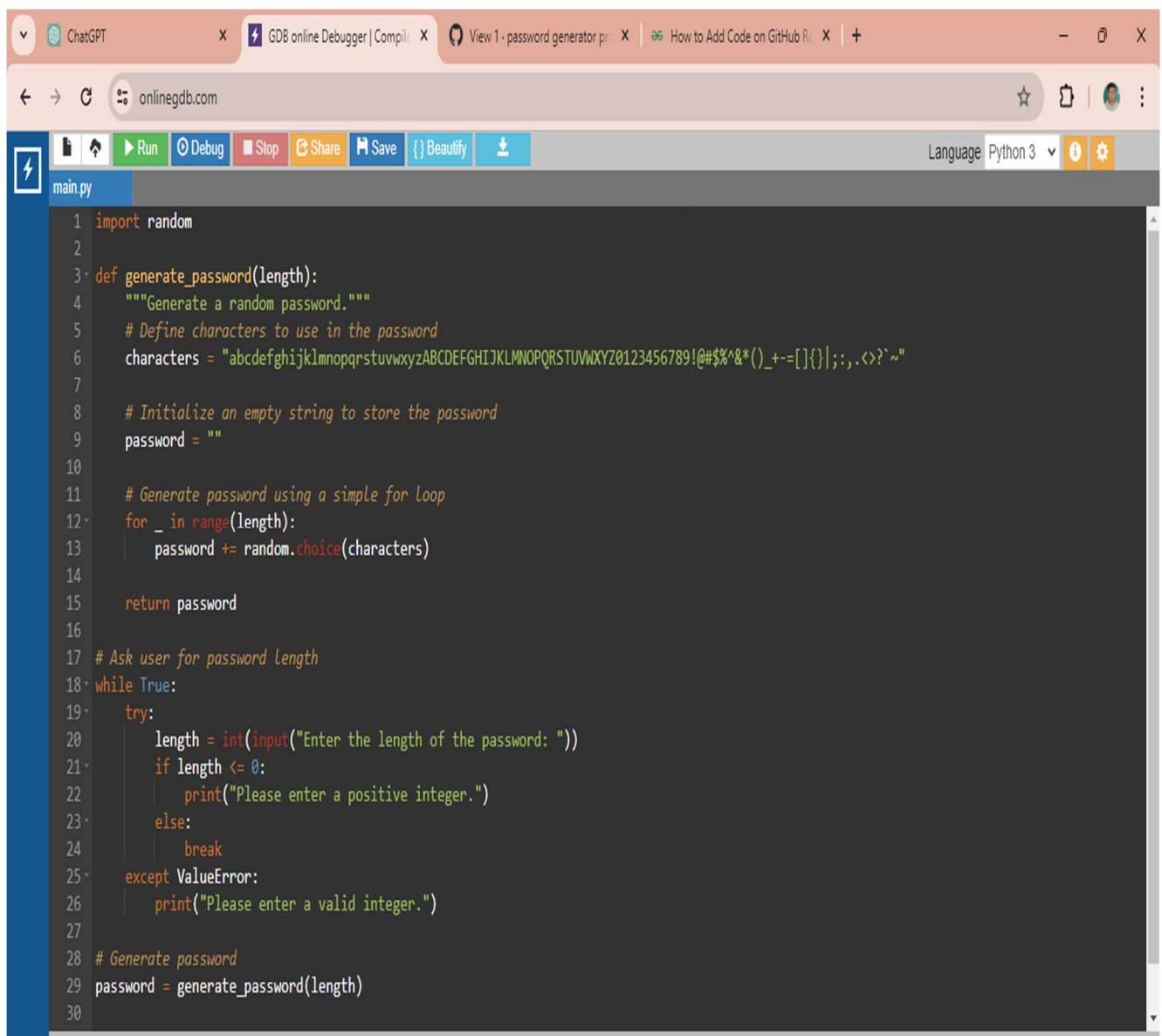
*print("Please enter a valid integer.")*

*# Generate password*

*password = generate\_password(length)*

*# Print generated password*

*print("Generated Password:", password)*

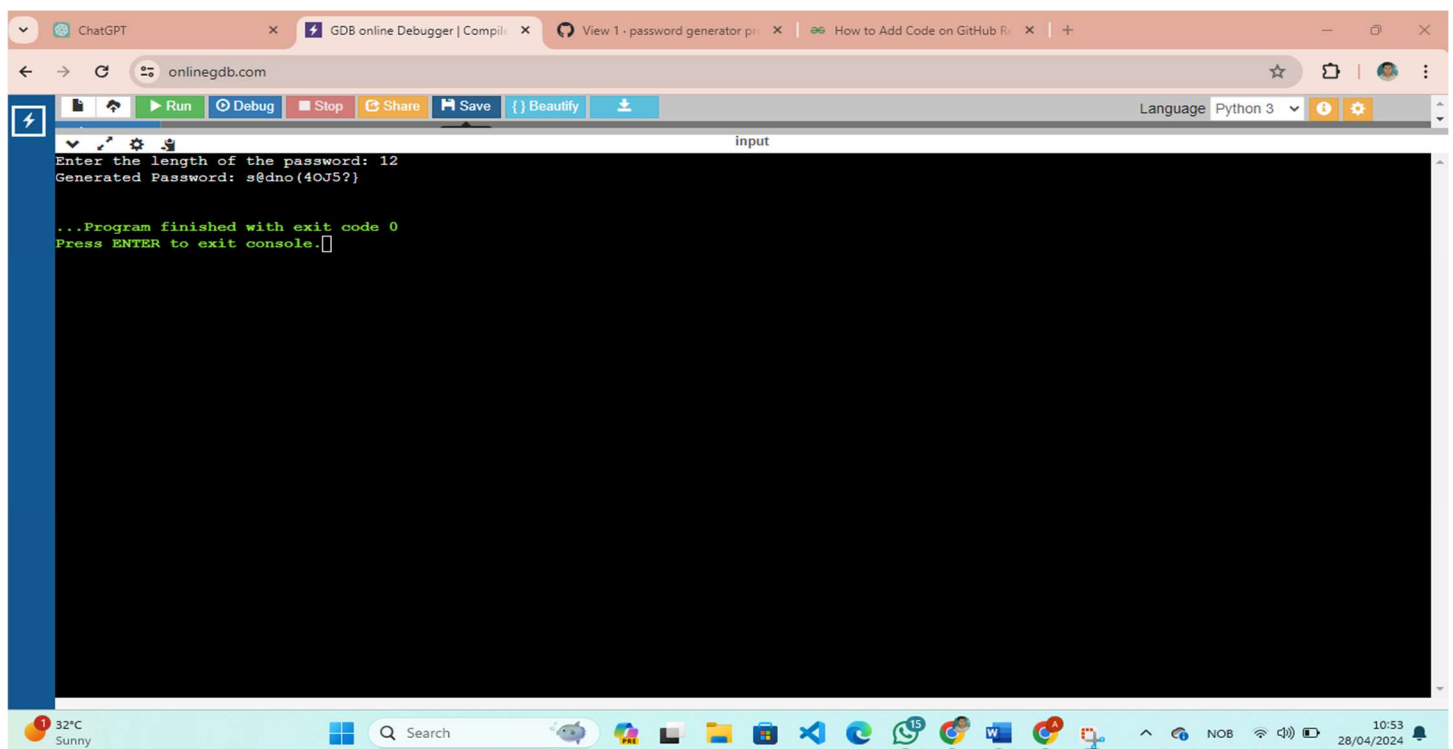


The screenshot shows a web browser window with the URL `onlinegdb.com`. The browser has several tabs open, including "ChatGPT", "GDB online Debugger | Compile", "View 1 - password generator pr...", and "How to Add Code on GitHub Re...". The IDE interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and Download. The language is set to Python 3. The code editor displays a Python script for a password generator. The script imports the `random` module and defines a `generate_password(length)` function. This function generates a random password of a specified length using a set of characters including letters, digits, and special characters. The main part of the script uses a `while True:` loop to repeatedly ask the user for a password length, validate it (ensuring it's a positive integer), and then generate and print the password.

```
1 import random
2
3 def generate_password(length):
4     """Generate a random password."""
5     # Define characters to use in the password
6     characters = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789!@#$%^&*()_+=[{}|;:,.<>?'`~"
7
8     # Initialize an empty string to store the password
9     password = ""
10
11     # Generate password using a simple for loop
12     for _ in range(length):
13         password += random.choice(characters)
14
15     return password
16
17 # Ask user for password length
18 while True:
19     try:
20         length = int(input("Enter the length of the password: "))
21         if length <= 0:
22             print("Please enter a positive integer.")
23         else:
24             break
25     except ValueError:
26         print("Please enter a valid integer.")
27
28 # Generate password
29 password = generate_password(length)
30
```

```
# Print generated password  
print("Generated Password:", password)
```

## OUTPUT



The screenshot shows a web browser window with several tabs open. The active tab is 'onlinegdb.com'. The browser's address bar shows 'onlinegdb.com'. Below the address bar, there is a toolbar with buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', and 'Beautify'. The main content area displays the output of a program. The text in the console is as follows:

```
input  
Enter the length of the password: 12  
Generated Password: s@dn0(40J5?}  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

The browser's taskbar at the bottom shows the system clock as 10:53 on 28/04/2024, and the weather as 32°C Sunny.