Ayoob Ibrahim, Branden Barnes, Caleb Mayo, Bilal Boumahdi

Professor Shobha Vatsa

7 December 2022

Group Project

Problem Statement:

A general problem in Cybersecurity are breaches due to weak or simple passwords within an infrastructure, by creating a random password generator with complexity requirements it would mitigate that problem.

Hardware:

Device name	DESKTOP-C08GMHG
Processor	Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz 2.11 GHz
Installed RAM	8.00 GB (7.79 GB usable)
Device ID	CB168264-D60F-4404-B5D6-F4D264ADFF72
Product ID	00325-81518-94768-AAOEM
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display

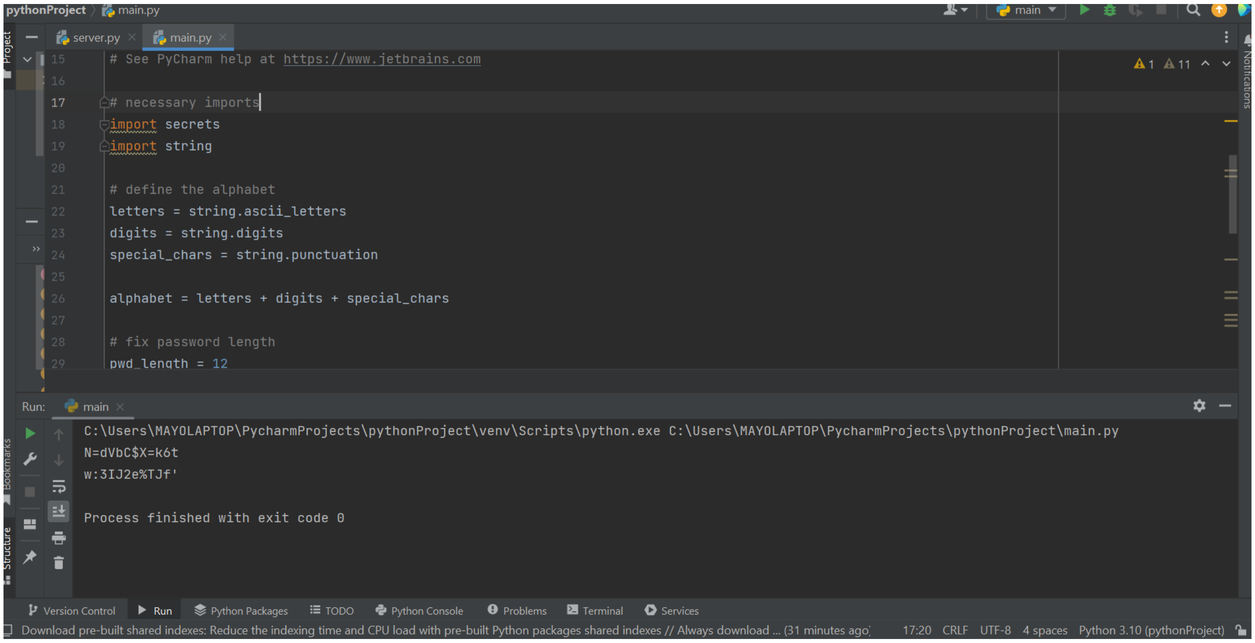
Edition	Windows 10 Home
Version	21H2
Installed on	4/1/2021
OS build	19044.2251
Experience	Windows Feature Experience Pack 120.2212.4180.0

Software:

Python 3.10 (Python Project)

PyCharm Community Edition 2022.2.3

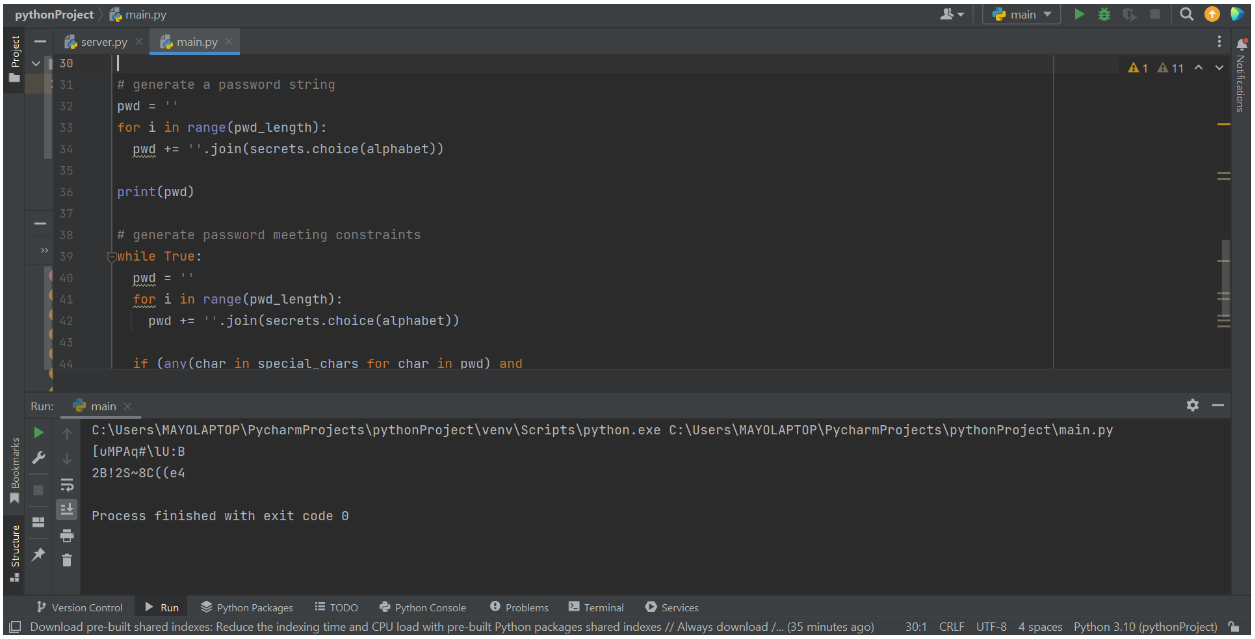
Screenshots:



```
# See PyCharm help at https://www.jetbrains.com

# necessary imports
import secrets
import string

# define the alphabet
letters = string.ascii_letters
digits = string.digits
special_chars = string.punctuation

alphabet = letters + digits + special_chars

# fix password length
pwd_length = 12
```

```
C:\Users\MAYOLAPTOP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\MAYOLAPTOP\PycharmProjects\pythonProject\main.py
N=dVbC$X=k6t
w:3IJ2e%TJf'

Process finished with exit code 0
```

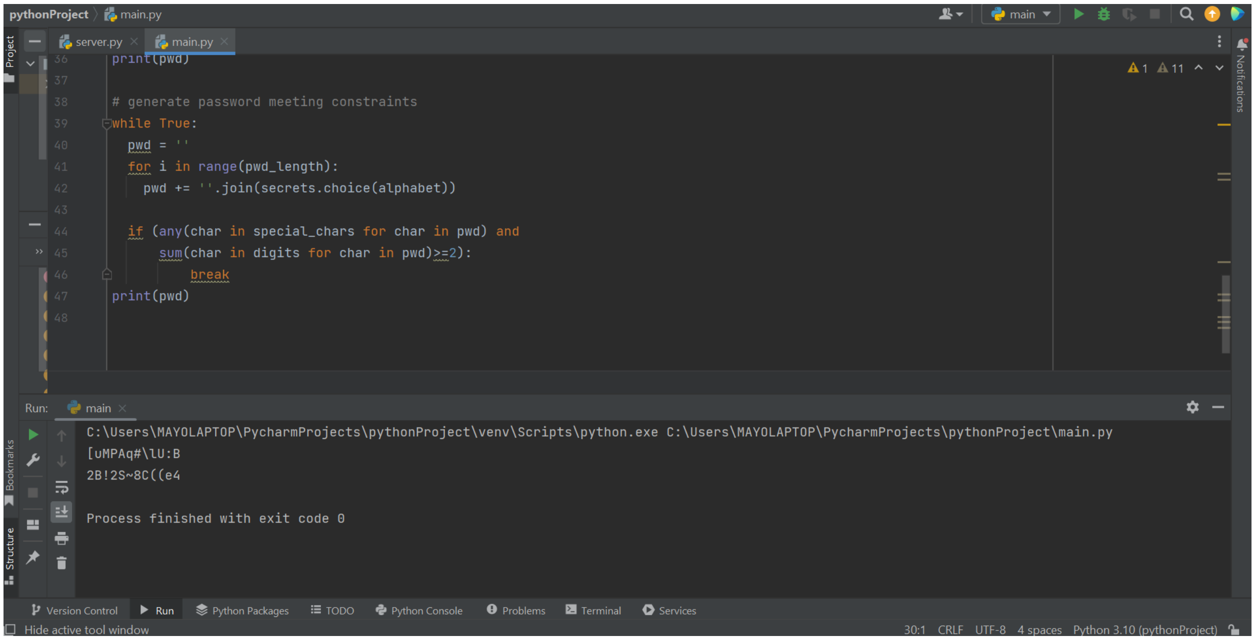
Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download ... (31 minutes ago)



```

# generate a password string
pwd = ''
for i in range(pwd_length):
  pwd += ''.join(secrets.choice(alphabet))

print(pwd)

# generate password meeting constraints
while True:
  pwd = ''
  for i in range(pwd_length):
    pwd += ''.join(secrets.choice(alphabet))

  if (any(char in special_chars for char in pwd) and
```

```
C:\Users\MAYOLAPTOP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\MAYOLAPTOP\PycharmProjects\pythonProject\main.py
[uMPAq#\lU:B
2B!2S~8C((e4

Process finished with exit code 0
```

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download /... (35 minutes ago)



```
print(pwd)

# generate password meeting constraints
while True:
  pwd = ''
  for i in range(pwd_length):
    pwd += ''.join(secrets.choice(alphabet))

  if (any(char in special_chars for char in pwd) and
      sum(char in digits for char in pwd)>=2):
          break
print(pwd)
```

```
C:\Users\MAYOLAPTOP\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\MAYOLAPTOP\PycharmProjects\pythonProject\main.py
[uMPAq#\lU:B
2B!2S~8C((e4

Process finished with exit code 0
```

Description:

First we imported the secrets module that creates a combination of numbers, letters, and special characters into a strong random combination of line or text to create passwords. Next we imported the string module to define the alphabet for the password generator. “ascii _letters” defines the lower and upper case letters, while the digits constant defines numbers 0 through 9. The punctuation constant defined the special characters. After defining those, we used the alphabet as our variable to bring all the constants together. Next, we set the password length to a fixed number, 12. We then set the password string to be empty or unclear. “Secrets.choice(alphabet)” outputs one character from the alphabet. The join() function adds the character from the secrets.choice(alphabet) to the password string. We then created an if statement that checks the code generated based on the conditions that we set. Last, we printed the pwd output.

References

C, B. P. (2022, September 26). How to create a random password generator in Python. Geekflare. Retrieved December 6, 2022, from https://geekflare.com/password-generator-python-code/

YouTube. (2021). *YouTube*. Retrieved December 6, 2022, from https://www.youtube.com/watch?v=_3dYO5fNcHI&ab_channel=ShriramVasudevan.