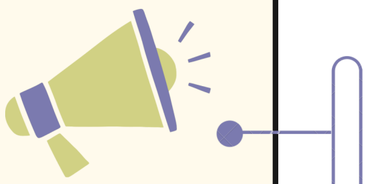
GROUP B

wireshark

COMPUTER NETWORKS

Cisco packet

- Taazim Toktogulov
- RAKHIMOV RUSTAMBEK ISMOILBEK UGLI



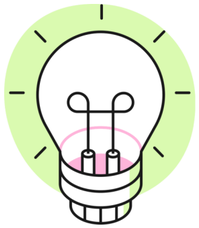
Ads

Python

TASKS

A CLIENT CAN SEND AND VISUALIZE THE IMAGE, CSV AND JSON FILES IN OTHER CLIENT SIDE

- A CLIENT CAN BROADCAST ANY FILE TO MULTIPLE CLIENTS.



- A MENU FORM SELECTING CHOICES

INTRODUCTION

Broadcast to client1 and client 2

CLIENT 1
 # importing necessary libraries
import socket
import struct

 # setting up the IP/PORT and buffer size
serverip = "224.1.1.1"
serverport = 6464
bufferSize = 2048

 # Setting up the connection by connecting our code to the socket
ClientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
ClientSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
ClientSocket.bind(('', serverport)) # Binding the socket to the ip and port

 # Code below is used to enable UDP multicast
var = struct.pack("4sl", socket.inet_aton(serverip), socket.INADDR_ANY)
ClientSocket.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, var)

 # Listen for messages from the server/sender and decoding it
msgFromServer = ClientSocket.recvfrom(bufferSize)
msg = "Message from Server {}".format(msgFromServer[0].decode())

 # Printing the message received for user view
print(msg)

CLIENT 2
 # importing necessary libraries
import socket
import struct

 # setting up the IP/PORT and buffer size
serverip = "224.1.1.1"
serverport = 6464
bufferSize = 2048

 # Setting up the connection by connecting our code to the socket
ClientSocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM, socket.IPPROTO_UDP)
ClientSocket.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
ClientSocket.bind(('', serverport)) # Binding the socket to the ip and port

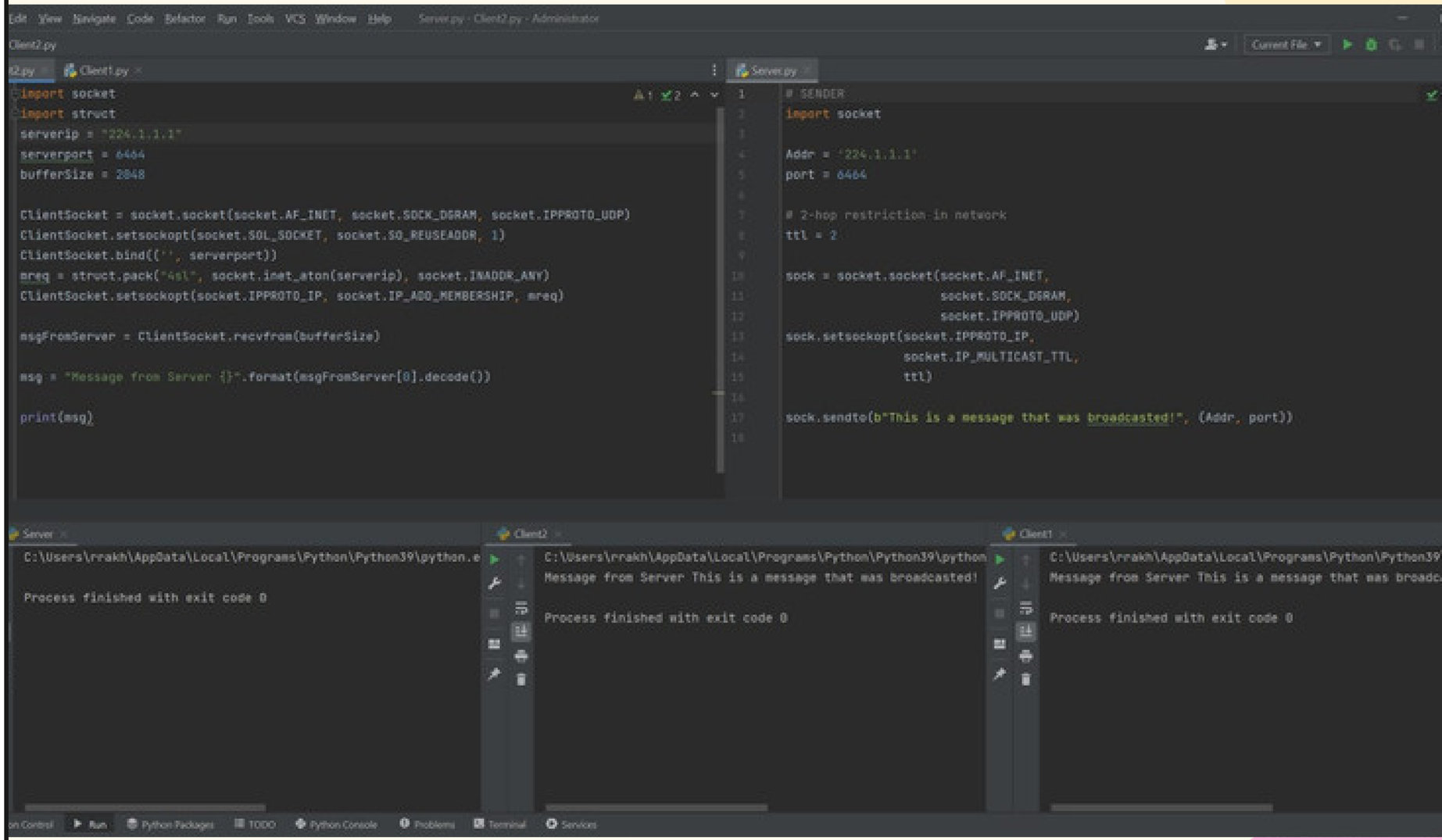
 # Code below is used to enable UDP multicast
var = struct.pack("4sl", socket.inet_aton(serverip), socket.INADDR_ANY)
ClientSocket.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, var)

 # Listen for messages from the server/sender and decode it
msgFromServer = ClientSocket.recvfrom(bufferSize)
msg = "Message from Server {}".format(msgFromServer[0].decode())

 # Printing the message received for user view
print(msg)

server

```
# SERVER
# IMPORTING THE SOCKET LIBRARY REQUIRED FOR THE SENDER/SERVER
import socket

# SETTING THE IP AND PORT USED FOR CREATING A SOCKET
Addr = '224.1.1.1'
port = 6464

# 2-hop restriction in network
ttl = 2

# Setting up the socket for a multicast that allows sending and receiving a broadcast.
sock = socket.socket(socket.AF_INET,
         socket.SOCK_DGRAM,
         socket.IPPROTO_UDP)
sock.setsockopt(socket.IPPROTO_IP,
       socket.IP_MULTICAST_TTL,
       ttl)
# The code below sends a broadcast to client 1 and client 2
sock.sendto(b"This is a message that was broadcasted!", (Addr, port))
```

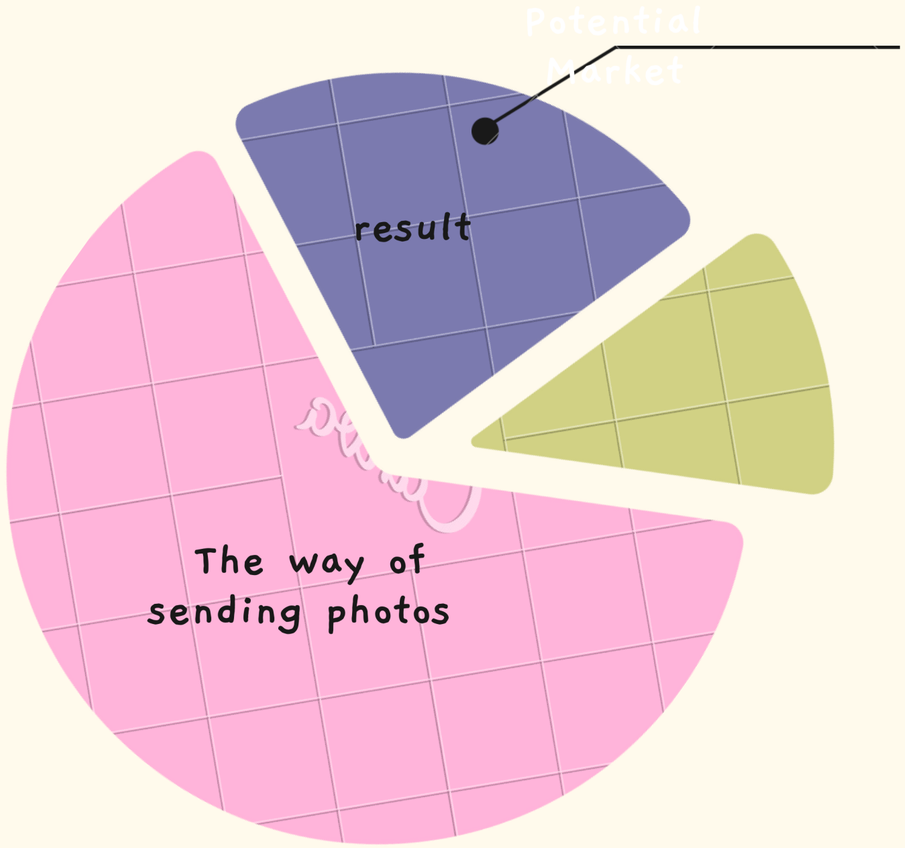


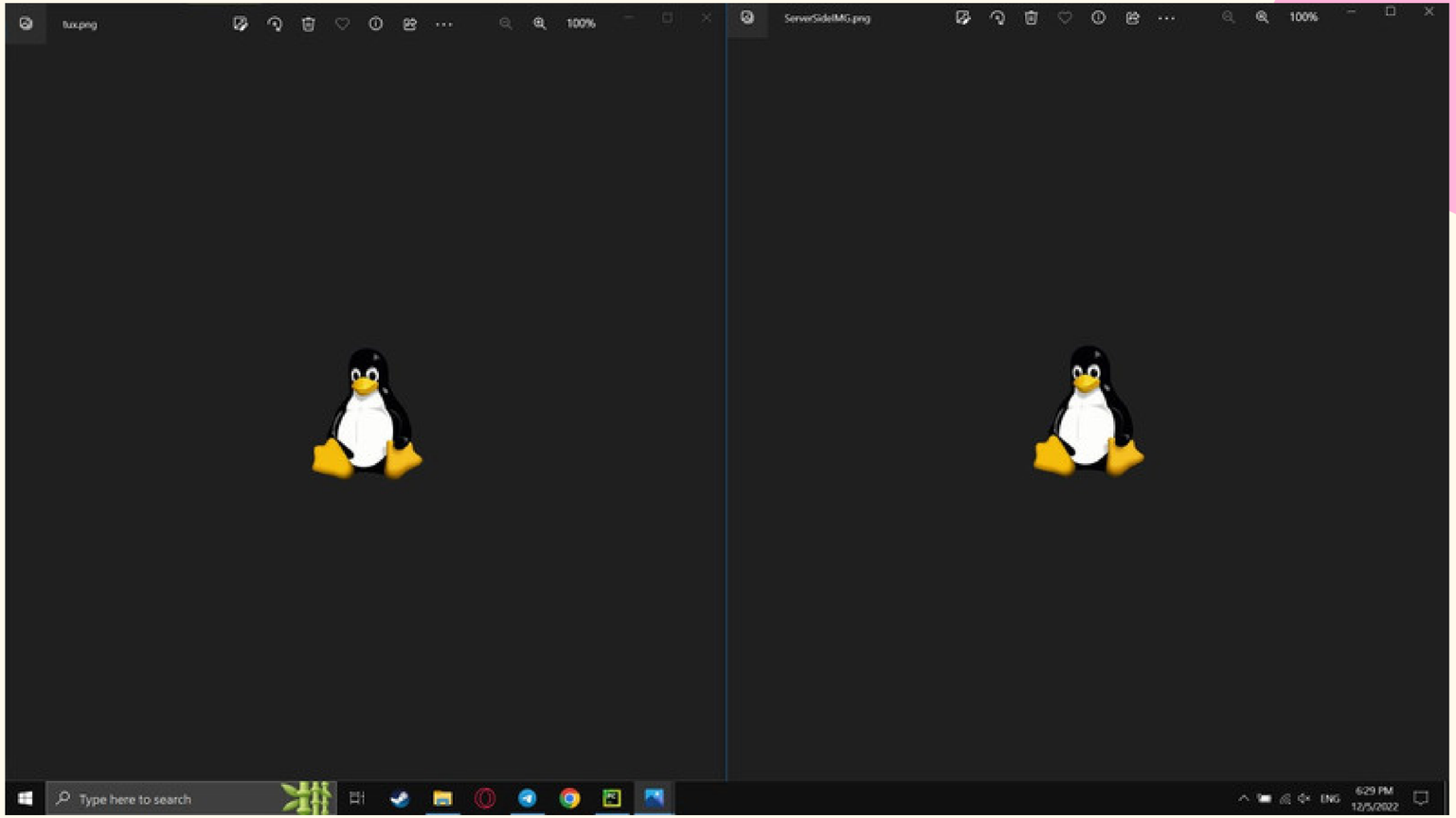
SENDING IMAGE, CSV FILE OR JSON FILE

```
def switch1(user_choice):
    if user_choice == "IMG":
        q = 0
        type_send = str.encode("IMG")   # Encoding the data type
        UDPClientSocket.sendto(type_send, serverAddressPort)    # Sending the data type
        time.sleep(1)   # Waiting for the server

        file = open('tux.png', 'rb')
        image = file.read(1024)
        while image:
            start = time.time()
            bytesToSend = str.encode(str(image))                    # Encoding the data line by line
            UDPClientSocket.sendto(bytesToSend, serverAddressPort)      # Sending the data line by line
            time.sleep(1)   # Timer used to calculate RTT
            msgFromServer = UDPClientSocket.recvfrom(bufferSize)        # Catching server response
            msg = "Message from Server {}".format(msgFromServer[0])     # Viewing server response

            if msg == "Message from Server b'Pong!'":   # If ACK was received:
                print("RTT: ", time.time() - start)     # Timer used to calculate RTT
                print("ACK from server received.")      # Inform the user that the ACK was received.
```

```
-IMG -> Image
-CSV -> Csv File
-JSON -> Json File
What would you like to send?:IMG
RTT:  1.011275291442871
ACK from server received.
RTT:  1.0057628154754639
ACK from server received.
RTT:  1.0044441223144531
ACK from server received.
RTT:  1.010587453842163
```

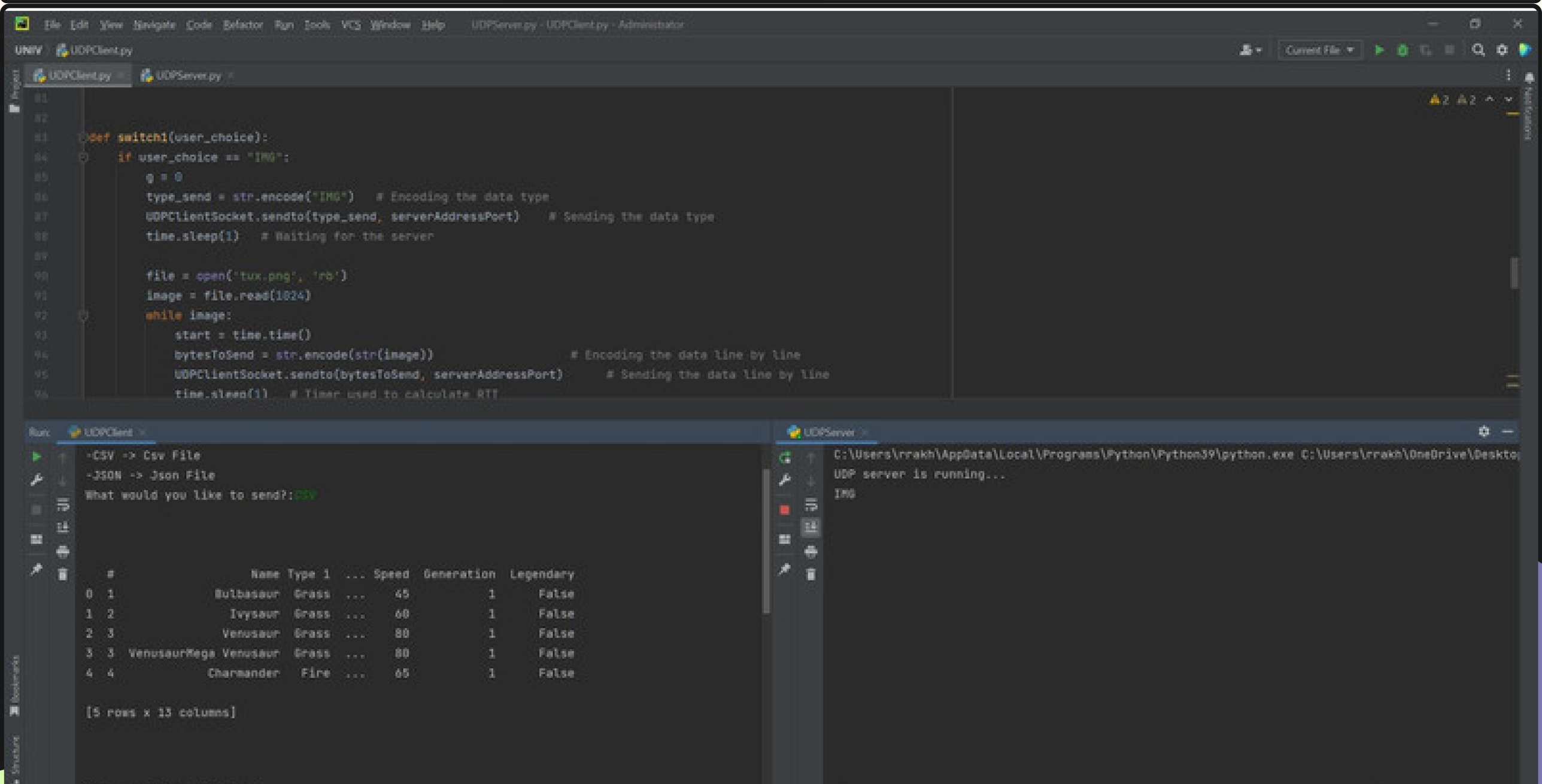
```
C:\Users\rrakh\AppData\Local\Programs\Python\Python39\python.exe C:\Users\rrakh\OneDrive\Deskto
UDP server is running...
IMG
```





This is output of Image

CSV FILE OUTPUT AND IT'S CODE

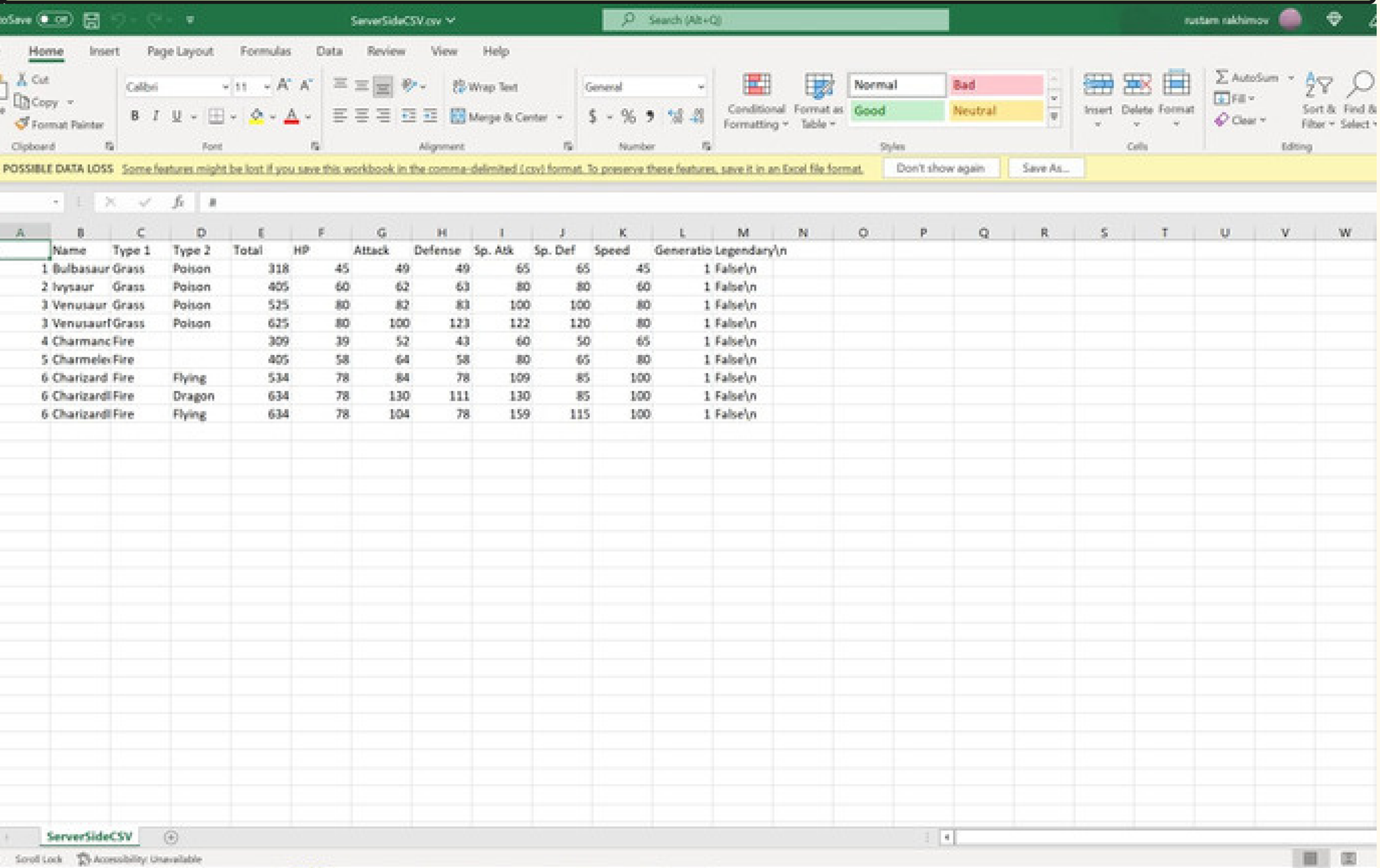


```
def switch1(user_choice):
    if user_choice == "IMG":
        q = 0
        type_send = str.encode("IMG")   # Encoding the data type
        UDPClientSocket.sendto(type_send, serverAddressPort)    # Sending the data type
        time.sleep(1)   # Waiting for the server

        file = open('tux.png', 'rb')
        image = file.read(1024)
        while image:
            start = time.time()
            bytesToSend = str.encode(str(image))                # Encoding the data line by line
            UDPClientSocket.sendto(bytesToSend, serverAddressPort)      # Sending the data line by line
            time.sleep(1)   # Timer used to calculate RTT
```

```
-CSV -> Csv File
-JSON -> Json File
What would you like to send?:CSV



   #                  Name Type 1  ... Speed  Generation  Legendary
0  1             Bulbasaur  Grass  ...    45           1      False
1  2               Ivysaur  Grass  ...    60           1      False
2  3              Venusaur  Grass  ...    80           1      False
3  3  VenusaurMega Venusaur  Grass  ...    80           1      False
4  4            Charmander   Fire  ...    65           1      False

[5 rows x 13 columns]
```

```
C:\Users\rrakh\AppData\Local\Programs\Python\Python39\python.exe C:\Users\rrakh\OneDrive\Deskto
UDP server is running...
IMG
```

CSV FILE OUTPUT



ServerSideCSV.csv

Search (Alt+Q)

rustam rakhimov

Home Insert Page Layout Formulas Data Review View Help

Cut Copy Format Painter — Clipboard

Calibri 11 — Font

Wrap Text Merge & Center — Alignment

General — Number

Conditional Formatting Format as Table — Styles

Normal Bad Good Neutral — Styles

Insert Delete Format — Cells

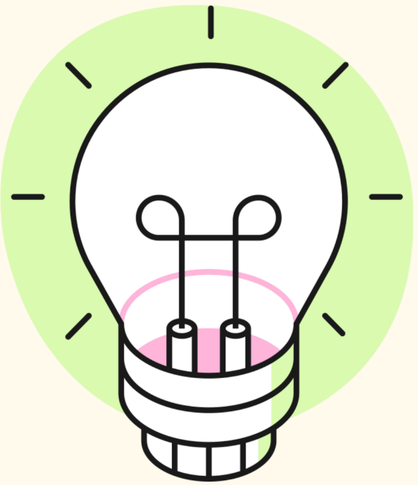
AutoSum Fill Clear Sort & Filter Find & Select — Editing

POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format. Don't show again Save As...

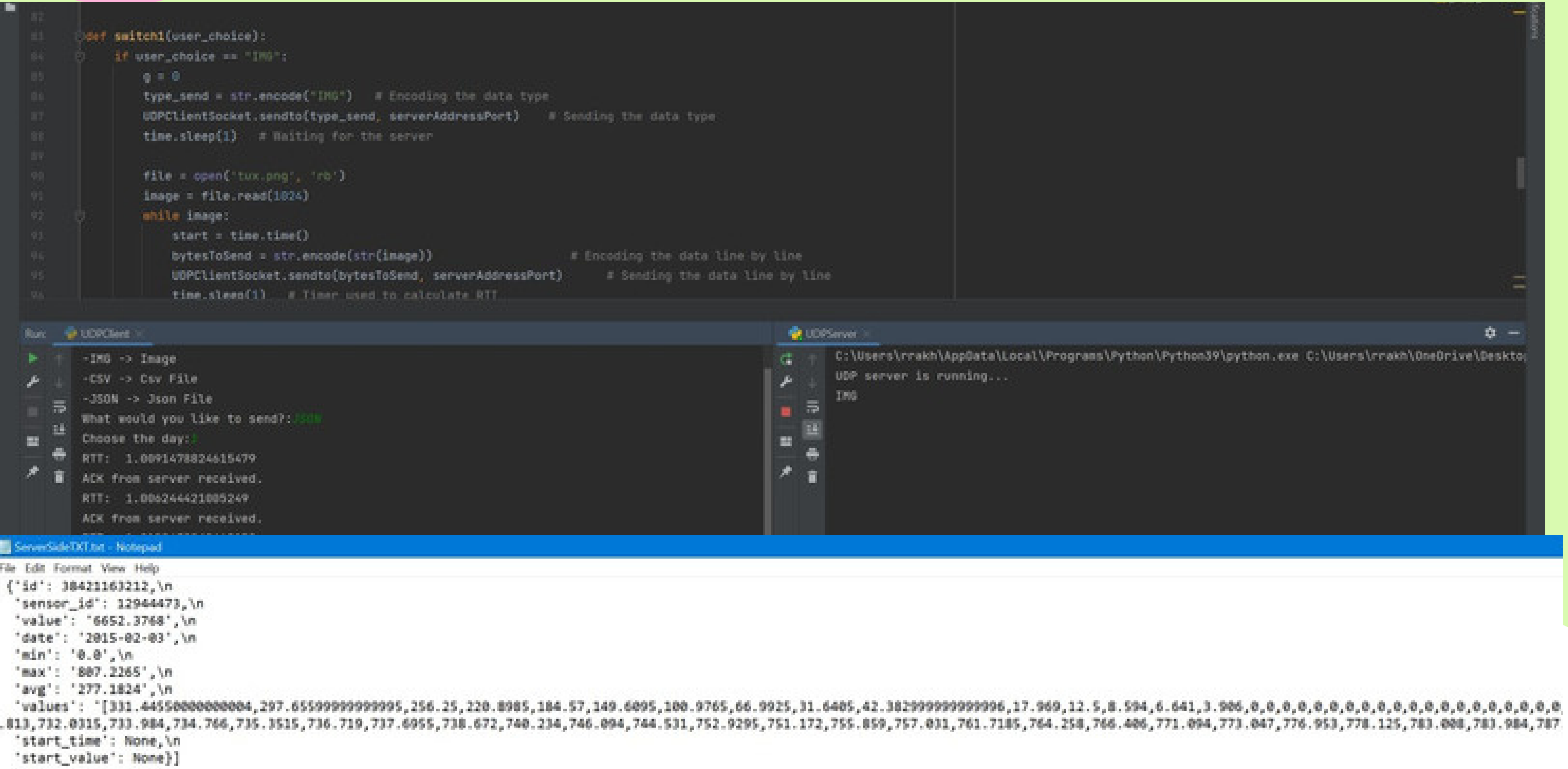
	Name	Type 1	Type 2	Total	HP	Attack	Defense	Sp. Atk	Sp. Def	Speed	Generatio	Legendary\n
1	Bulbasaur	Grass	Poison	318	45	49	49	65	65	45	1	False\n
2	Ivysaur	Grass	Poison	405	60	62	63	80	80	60	1	False\n
3	Venusaur	Grass	Poison	525	80	82	83	100	100	80	1	False\n
3	VenusaurM	Grass	Poison	625	80	100	123	122	120	80	1	False\n
4	Charmanc	Fire		309	39	52	43	60	50	65	1	False\n
5	Charmele	Fire		405	58	64	58	80	65	80	1	False\n
6	Charizard	Fire	Flying	534	78	84	78	109	85	100	1	False\n
6	CharizardM	Fire	Dragon	634	78	130	111	130	85	100	1	False\n
6	CharizardM	Fire	Flying	634	78	104	78	159	115	100	1	False\n

ServerSideCSV

Scroll Lock Accessibility: Unavailable



JSON FILE CODE AND OUTPUT



COMPUTER NETWORKS

THANKS PROFESSOR

for a wonderful semester. I appreciated your caring disposition and teaching style. Thanks for your attitude and great job!!!

THANKS, GROUPMATES, AND TEAMMATES

For hard-working and desire!
Good luck on the semester :)