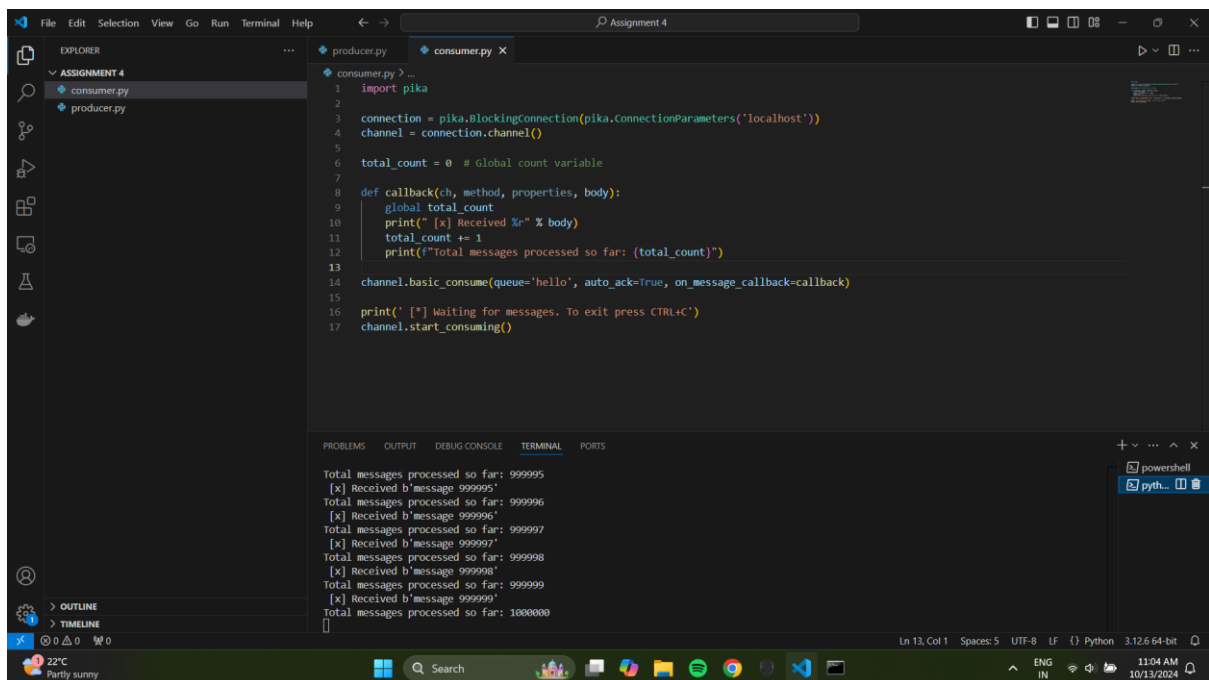


Message Queues: RabbitMQ

To create a simple messaging system, I used Producer and Consumer applications, paired with RabbitMQ. The producer loops a million times to post a message to the queue, and the consumer consumes the messages from the queue. The code for this is available on [Github](#).

After installing and starting RabbitMQ on my system, I started both applications together so the consumer started reading from the queue in real-time. After the producer application had sent a million messages and consumer had received them, I was able to confirm the same from the logs by maintaining a global variable 'total count' in the consumer application. It was observed that all 1,000,000 messages were consumed correctly.



The screenshot shows a VS Code editor with a file explorer on the left showing 'ASSIGNMENT 4' containing 'consumer.py' and 'producer.py'. The main editor displays 'consumer.py' with the following code:

```
1 import pika
2
3 connection = pika.BlockingConnection(pika.ConnectionParameters('localhost'))
4 channel = connection.channel()
5
6 total_count = 0 # Global count variable
7
8 def callback(ch, method, properties, body):
9     global total_count
10    print("[x] Received %r" % body)
11    total_count += 1
12    print(f"Total messages processed so far: {total_count}")
13
14 channel.basic_consume(queue='hello', auto_ack=True, on_message_callback=callback)
15
16 print("[*] Waiting for messages. To exit press CTRL+C")
17 channel.start_consuming()
```

The terminal at the bottom shows the output of the consumer application:

```
Total messages processed so far: 999995
[x] Received b'message 999995'
Total messages processed so far: 999996
[x] Received b'message 999996'
Total messages processed so far: 999997
[x] Received b'message 999997'
Total messages processed so far: 999998
[x] Received b'message 999998'
Total messages processed so far: 999999
[x] Received b'message 999999'
Total messages processed so far: 1000000
```

Having enabled the RabbitMQ Management Plugin, I was able to view some additional metadata on the 15672 port on localhost. It was observed that the consumption of the messages was entirely real-time as it showed no lags on the graphs. Through these two, we can confidently claim that not a single message was dropped in the queue and they were successfully consumed.

