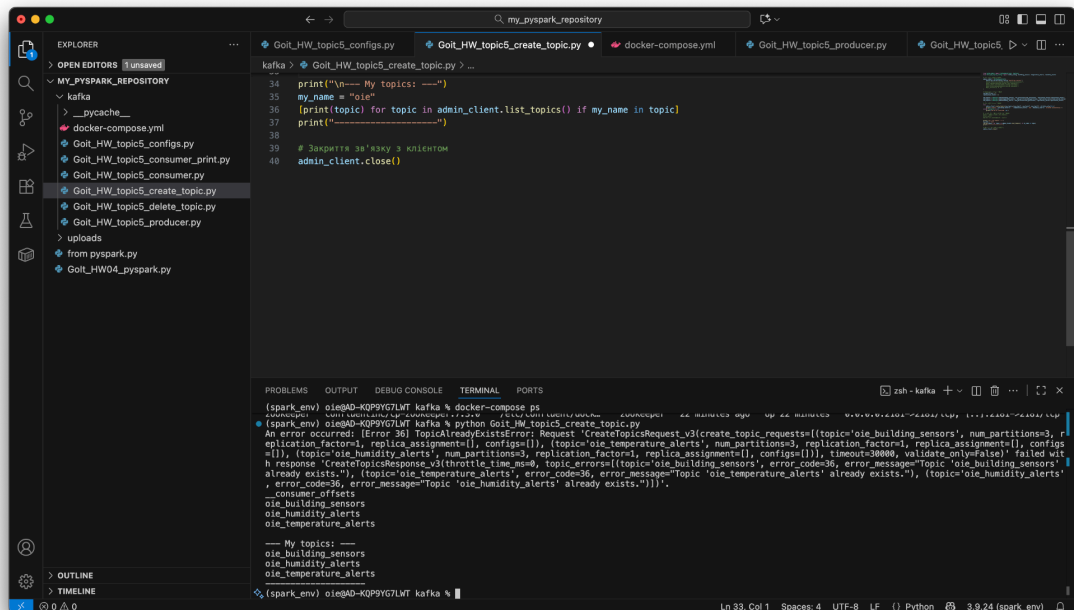


Кроки виконання завдання:

1. Створення topics. Було створено три топіка: oie_building_sensors, oie_humidity_alerts, oie_temperature_alerts на локальному сервері. Для цього створено контейнер kafka через файл docker-compose.yml

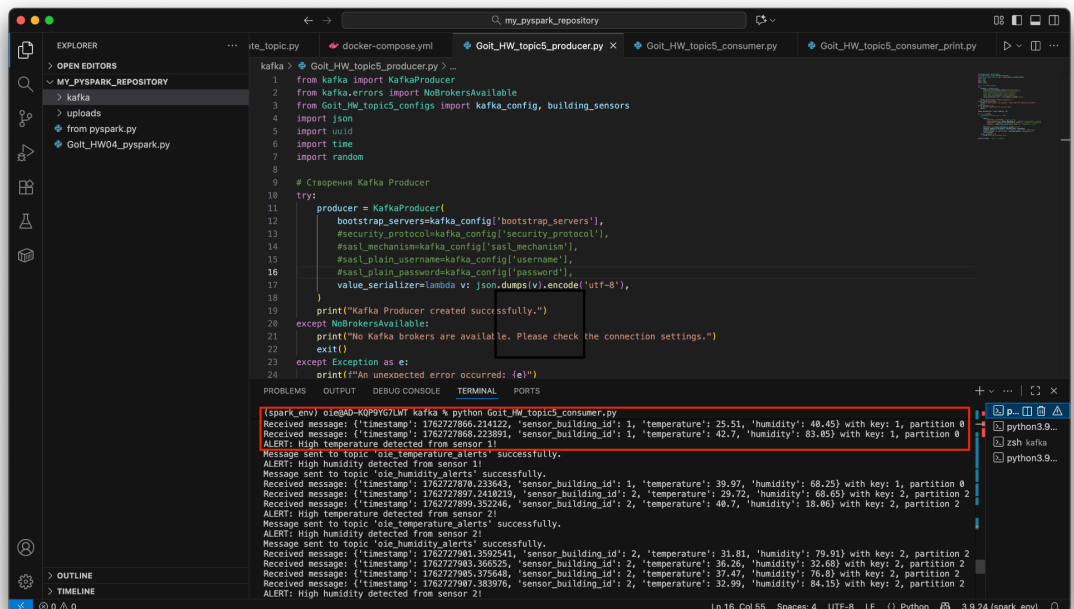


```
34 print("\n--- My topics: ---")
35 my_name = "oie"
36 (print(topic for topic in admin_client.list_topics() if my_name in topic)
37 print("-----"))
38
39 # Записати записи в клієнт
40 admin_client.close()
```

```
(spark_env) oie@40-K0P9YG7LW7 kafka % docker-compose ps
(spark_env) oie@40-K0P9YG7LW7 kafka % python Goit_HW_topic5_create_topic.py
An error occurred: (Error 36) TopicAlreadyExistsError: Request 'createtopicrequests_v3(create_topic_requests=[(topic:'oie_building_sensors', num_partitions=3, replication_factor=1, replica_assignment=[], configs=[]), (topic:'oie_humidity_alerts', num_partitions=3, replication_factor=1, replica_assignment=[], configs=[]), (topic:'oie_temperature_alerts', num_partitions=3, replication_factor=1, replica_assignment=[], configs=[])], timeout=30000, validate_only=False)' failed with response 'CreateTopicsResponse_v3(brotli_time_ms=4, topic_errors=[(topic:'oie_building_sensors', error_code=36, error_message='Topic 'oie_building_sensors' already exists.'), (topic:'oie_temperature_alerts', error_code=36, error_message='Topic 'oie_temperature_alerts' already exists.'), (topic:'oie_humidity_alerts', error_code=36, error_message='Topic 'oie_humidity_alerts' already exists.')])'.
```

```
--- My topics: ---
oie_building_sensors
oie_humidity_alerts
oie_temperature_alerts
```

2. Створено та запущено в роботу consumer (підписаний на oie_building_sensors). Що приймає сповіщення від provider з даними, отриманими від датчиків (у даному випадку два датчика sensor_building_1, sensor_building_2) та обробляє отримані дані.



```
1 from kafka import KafkaProducer
2 from kafka.errors import NoBrokersAvailable
3 from Goit_HW_topic5_configs import kafka_config, building_sensors
4 import json
5 import uuid
6 import time
7 import random
8
9 # Створення Kafka Producer
10 try:
11     producer = KafkaProducer(
12         bootstrap_servers=kafka_config['bootstrap_servers'],
13         security_protocol=kafka_config['security_protocol'],
14         sasl_mechanism=kafka_config['sasl_mechanism'],
15         sasl_plain_username=kafka_config['username'],
16         sasl_plain_password=kafka_config['password'],
17         value_serializer=lambda v: json.dumps(v).encode('utf-8'),
18     )
19     print("Kafka Producer created successfully.")
20 except NoBrokersAvailable:
21     print("No Kafka brokers are available. Please check the connection settings.")
22     exit()
23 except Exception as e:
24     print(f"An unexpected error occurred: {e}")
```

```
(spark_env) oie@40-K0P9YG7LW7 kafka % python Goit_HW_topic5_consumer.py
Received message: {'timestamp': 1762727866.214322, 'sensor_building_id': 1, 'temperature': 25.51, 'humidity': 40.45} with key: 1, partition 0
Received message: {'timestamp': 1762727866.223891, 'sensor_building_id': 1, 'temperature': 42.7, 'humidity': 83.85} with key: 1, partition 0
ALERT: High temperature detected from sensor 1!
Message sent to topic 'oie_temperature_alerts' successfully.
ALERT: High humidity detected from sensor 1!
Message sent to topic 'oie_humidity_alerts' successfully.
Received message: {'timestamp': 1762727870.233643, 'sensor_building_id': 1, 'temperature': 39.97, 'humidity': 68.25} with key: 1, partition 0
Received message: {'timestamp': 1762727870.241029, 'sensor_building_id': 2, 'temperature': 29.72, 'humidity': 66.65} with key: 2, partition 2
Received message: {'timestamp': 1762727893.352246, 'sensor_building_id': 2, 'temperature': 48.7, 'humidity': 38.80} with key: 2, partition 2
ALERT: High temperature detected from sensor 2!
Message sent to topic 'oie_temperature_alerts' successfully.
ALERT: High humidity detected from sensor 2!
Message sent to topic 'oie_humidity_alerts' successfully.
Received message: {'timestamp': 1762727903.366525, 'sensor_building_id': 2, 'temperature': 31.81, 'humidity': 79.91} with key: 2, partition 2
Received message: {'timestamp': 1762727905.375646, 'sensor_building_id': 2, 'temperature': 36.26, 'humidity': 32.68} with key: 2, partition 2
Received message: {'timestamp': 1762727905.375646, 'sensor_building_id': 2, 'temperature': 37.47, 'humidity': 76.8} with key: 2, partition 2
Received message: {'timestamp': 1762727907.383976, 'sensor_building_id': 2, 'temperature': 32.99, 'humidity': 84.15} with key: 2, partition 2
ALERT: High humidity detected from sensor 2!
```

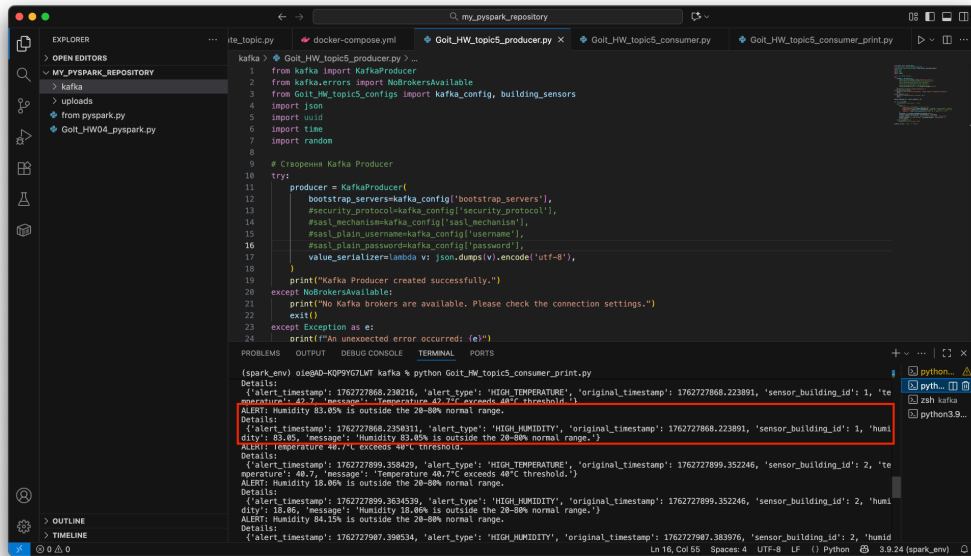
У разі відповідності параметрів заданим умовам (перевищення певного порогу значень) вбудований provider формує alert сповіщення та відправляє його в інші топіки (oie_humidity_alerts, oie_temperature_alerts) в залежності від того, який з параметрів відповідає заданим умовам.

The screenshot shows a VS Code editor with a file explorer on the left, a code editor in the center, and a terminal at the bottom. The code editor displays a Python script named `Goit_HW_topic5_producer.py` that uses the `kafka-python` library to connect to a Kafka broker and send messages to the `oie_building_sensors` topic. The script includes error handling for `NoBrokersAvailable` and general exceptions. The terminal at the bottom shows the output of the command `python Goit_HW_topic5_consumer.py` in a `spark_env` environment. It displays received messages from the `oie_building_sensors` topic, including timestamps, sensor IDs, and temperature/humidity data. Alerts for high temperature and high humidity are also shown.

3. Створено та запущено в роботу provider. Що імітує роботу двох датчиків (у даному випадку два датчика `sensor_building_1`, `sensor_building_2`), генерує сповіщення з даними, отриманими від датчиків, та відправляє сповіщення в топик `oie_building_sensors`, який слухає consumer (підписаний на `oie_building_sensors`).

This screenshot is identical to the one above, showing the same VS Code editor setup. It displays the same Python script for the Kafka producer and the same terminal output showing messages received by the consumer. The terminal output includes timestamps, sensor IDs, and temperature/humidity data, along with alerts for high temperature and high humidity.

4. Створено та запущено в роботу ще один consumer (підписаний на `oie_humidity_alerts`, `oie_temperature_alerts`), що приймає сповіщення від provider з alert сповіщеннями (містять оригінальні показники датчика), та виводять попередження у консоль, що містить показники датчика та причину сповіщення.



```
kafka > Golt_HW_topic5_producer.py > ...
1 from kafka import KafkaProducer
2 from kafka.errors import NoBrokersAvailable
3 from Golt_HW_topic5_configs import kafka_config, building_sensors
4 import json
5 import uuid
6 import time
7 import random
8
9 # Create a Kafka Producer
10 try:
11     producer = KafkaProducer(
12         bootstrap_servers=kafka_config['bootstrap_servers'],
13         security_protocol=kafka_config['security_protocol'],
14         sasl_mechanism=kafka_config['sasl_mechanism'],
15         sasl_plain_username=kafka_config['sasl_plain_username'],
16         sasl_plain_password=kafka_config['sasl_plain_password'],
17         value_serializer=lambda v: json.dumps(v).encode('utf-8'),
18     )
19     print("Kafka Producer created successfully.")
20 except NoBrokersAvailable:
21     print("No Kafka brokers are available. Please check the connection settings.")
22     exit()
23 except Exception as e:
24     print(f"An unexpected error occurred: {e}")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
(spark_env) ois@QD-KQPYGJLMT kafka % python Golt_HW_topic5_consumer_print.py
Details:
{'alert_timestamp': 1762727868.238216, 'alert_type': 'HIGH_TEMPERATURE', 'original_timestamp': 1762727868.223891, 'sensor_building_id': 1, 'temperature': 42.7, 'message': 'Temperature 42.7°C exceeds 40°C threshold.'}
ALERT: Humidity 83.85% is outside the 20-80% normal range.
Details:
{'alert_timestamp': 1762727868.238311, 'alert_type': 'HIGH_HUMIDITY', 'original_timestamp': 1762727868.223891, 'sensor_building_id': 1, 'humidity': 83.85, 'message': 'Humidity 83.85% is outside the 20-80% normal range.'}
ALERT: Temperature 40.7°C exceeds 40°C threshold.
Details:
{'alert_timestamp': 1762727899.358429, 'alert_type': 'HIGH_TEMPERATURE', 'original_timestamp': 1762727899.352246, 'sensor_building_id': 2, 'temperature': 40.7, 'message': 'Temperature 40.7°C exceeds 40°C threshold.'}
ALERT: Humidity 18.86% is outside the 20-80% normal range.
Details:
{'alert_timestamp': 1762727899.363439, 'alert_type': 'HIGH_HUMIDITY', 'original_timestamp': 1762727899.352246, 'sensor_building_id': 2, 'humidity': 18.86, 'message': 'Humidity 18.86% is outside the 20-80% normal range.'}
ALERT: Humidity 84.15% is outside the 20-80% normal range.
Details:
{'alert_timestamp': 1762727987.388534, 'alert_type': 'HIGH_HUMIDITY', 'original_timestamp': 1762727987.383976, 'sensor_building_id': 2, 'humidity': 84.15, 'message': 'Humidity 84.15% is outside the 20-80% normal range.'}
```