**1) Структура проєкту на Raspberry Pi**

/home/kateryna/myshop

├─ .env # змінні підключення до БД

├─ python/

│ ├─ db.py # функція get\_conn() читає .env

│ ├─ sale.py # генератор продажів + красивий лог

│ ├─ purchase.py # генератор закупівель + красивий лог

│ └─ test\_db.py / test\_connection.py (службові)

├─ sql/ # твої SQL-запити (якщо будуть)

└─ logs/

├─ sale.log # лог запусків sale.py (cron і ручні)

└─ purchase.log # лог запусків purchase.py

**2) Файл .env (на Pi)**

Розташування: /home/kateryna/myshop/.env  
Вміст (приклад — як у тебе):

DB\_HOST=127.0.0.1

DB\_USER=kateryna

DB\_PASSWORD=Savelieva7836

DB\_NAME=myshopdb

.env не у Git. db.py підтягує ці дані і підключається до MariaDB/MySQL.

**3) Налаштування cron (автозапуск кожні 30 хв)**

Переглянути:

crontab -l

Має бути:

SHELL=/bin/bash

\*/30 \* \* \* \* cd /home/kateryna/myshop/python && /usr/bin/python3 sale.py >> /home/kateryna/myshop/logs/sale.log 2>&1

\*/30 \* \* \* \* cd /home/kateryna/myshop/python && /usr/bin/python3 purchase.py >> /home/kateryna/myshop/logs/purchase.log 2>&1

Сервіс cron активний:

systemctl is-active cron # має показати: active

**4) Що роблять скрипти**

* sale.py: створює 1 продаж (випадковий клієнт, 1–3 товари з наявних), друкує:
  + роздільник ====
  + рядок старту з часом
  + повідомлення про створений продаж
  + рядок “✅ Verkauf generiert um …”
* purchase.py: аналогічно для закупівлі, завершує “✅ Einkauf generiert um …”.

Усе, що вони друкують, потрапляє у відповідні логи завдяки перенаправленню >> …log 2>&1 у cron.

**5) Де дивитися результати**

Логи на Pi:

tail -n 40 ~/myshop/logs/sale.log

tail -n 40 ~/myshop/logs/purchase.log

# live-режим:

tail -f ~/myshop/logs/sale.log

Перевірити, що записи в БД ростуть:

mysql -u kateryna -p -D myshopdb -e "SELECT COUNT(\*) FROM verkauf;"

mysql -u kateryna -p -D myshopdb -e "SELECT COUNT(\*) FROM einkauf;"

**6) Як ми це розгорнули (хронологія)**

1. Вирівняли Git на ноуті та Pi (git pull origin main).
2. Виправили структуру: перенесли файли з python/python у python/.
3. На Pi створили/поклали .env, змінили DB\_HOST на 127.0.0.1.
4. Переконалися, що db.py читає .env і конектиться до БД.
5. Додали в sale.py та purchase.py роздільники й часові логи.
6. Налаштували cron на 30 хв і перевірили, що сервіс активний.
7. Протестували ручні запуски та запис у logs/.

**7) Типові корисні команди**

# SSH з ноутбука:

ssh kateryna@192.168.31.237

# Шлях до Python на Pi:

which python3 # зазвичай /usr/bin/python3

# Ручний запуск як у cron (і створення логу):

cd ~/myshop/python && /usr/bin/python3 sale.py >> ~/myshop/logs/sale.log 2>&1

# Перегляд логів:

ls -l ~/myshop/logs

tail -n 50 ~/myshop/logs/sale.log

# Оновити код із GitHub на Pi:

cd ~/myshop && git pull origin main --rebase

**8) Де що НЕ з’явиться**

* На ноуті у папці myshop\logs\ може бути порожньо — логи створюються **на Raspberry Pi**, а не у Windows і не комітяться у Git.

**9) Якщо щось піде не так (швидкий чек-лист)**

* cron не пише логи → перевірити crontab -l і systemctl is-active cron.
* Лог порожній → запустити скрипт руками (команда з розділу 7) і подивитись помилку.
* Помилка доступу до БД → перевірити .env, спробувати mysql -u kateryna -p -D myshopdb -e "SHOW TABLES;".
* Немає .env → створити в /home/kateryna/myshop/.env (див. розділ 2).

**10) Наступні кроки (за бажанням)**

* Додати ротацію логів (щоб файли не росли безмежно).
* Написати кілька SQL-звітів у sql/ і зробити маленький run\_reports.py, який зберігає CSV.
* Окрема гілка feature/… для майбутніх змін і PR у main.

Якщо хочеш, зроблю коротку PDF-версію цієї інструкції або README-розділ у репозиторії.