-----Part B — Applied Scenario (Payment Flow)

1. **Key Steps**
    - **Checkout & Create Order**
        - User clicking a checkout in frontend.
        - Frontend send data for checkout to backend by POST /checkout.
        - Backend receive data from frontend and calculate total amount then save to database but mark status PENDING .
    - **Payment Request**
        - Backend prepare data for payment and Encode data to JWT.
        - Then Backend send API to Gateway to request token and redirect_url.
    - **User Redirect**
        - Backend send URL to Frontend then redirect user to Payment Gateway.
    - **Backend Return URL**
        - When payment is complete gateway will send request to url that we set in backend return url and data that receive from gateway will be Encode too.
    - **State Update**
        - Backend will Decode JWT to check status if success or failed then Update the order status to PAID or FAILED.
2. **Handling Failures & Reliability**
    - **Idempotency**
        - Use ID that can’t be duplicate so gateway can reject repeat payment.
    - **Timeout Rules**
        - Check if Order status is PENDING for example 60 minute change to EXPIRED.
    - **Reconciliation idea**
        - Use Polling to keep checking status for example every 5 or 10 minute.
3. **Security Considerations**
    - **JWT Integrity**
        - Using JWT is to ensure that data is not modified along the way because if JWT changed it will not match the secret key.
    - **No Sensitive Data**
        - Card numbers are not stored because user enter directly on Gateway page.
4. **What to monitor**
    - **Polling Rate**
        - Check The number of times we have to send API call to check status.
    - **Decryption Rate**
        - Number of times JWT decoding failed.
    - **Gateway Latency**
        - Measure the Gateway API response time.