

Session 13 – Payment Integration Lab (Stripe, PayPal, VNPay – Sandbox)

1. Nội dung

Context:

- Bạn đã có hệ thống:
 - Products, Cart (frontend).
 - Orders & Order Items (backend + frontend, Session 12).
 - Basic Admin order management.

Goal of this lab:

- Thêm layer thanh toán với **3 cổng sandbox**:
 - Stripe Checkout (Test Mode).
 - PayPal Sandbox (developers).
 - VNPay Sandbox (sandbox endpoints).
- Mỗi cổng sẽ:
 - Tạo một “payment session” từ một order.
 - Redirect user đến trang thanh toán của cổng (Stripe / PayPal / VNPay).
 - Sau khi thanh toán, redirect về trang **success** / **cancel** trên app của bạn.

2. Learning Objectives

Sau khi hoàn thành LAB, sinh viên có thể:

- Giải thích common payment flow:
 - Create payment session → redirect → callback → update order status.
- Tạo API backend để tích hợp 3 cổng thanh toán trong môi trường sandbox.
- Thiết kế frontend Payment Page cho phép:
 - Chọn cổng thanh toán.
 - Gọi API để tạo session và redirect.

- Xử lý các URL `success` / `cancel` để hiển thị kết quả và cập nhật trạng thái order.

3. High-Level Architecture & Flow

Cho SV vẽ sơ đồ và implement đúng theo:

1. Customer tạo order (Session 12).
2. Customer mở trang `Order Payment` (vd: `/orders/:id/pay`).
3. Chọn cổng: Stripe / PayPal / VNPAY.
4. Frontend gọi endpoint tương ứng:
 - `POST /payments/stripe/create-session`
 - `POST /payments/paypal/create-session`
 - `POST /payments/vnpay/create-session`
5. Backend tạo session với cổng (using sandbox credentials) và trả về **redirect URL**.
6. Frontend `window.location.href = redirectUrl`.
7. Sau khi thanh toán:
 - Cổng redirect về `success_url` hoặc `cancel_url` (trên frontend).
 - Đồng thời (tùy mức độ nâng cao) dùng webhook để backend cập nhật `order.status = PAID`.

Để tránh quá phức tạp, bạn có thể:

- Xử lý **redirect success/cancel** trên **frontend**,
- Backend có endpoint simple để “confirm payment” (mock webhook).

4. Backend – Setup Sandbox Credentials (Preparation)

4.1 Stripe Test Mode

Hướng dẫn trong đề bài (không cần code, chỉ yêu cầu SV tự tạo account):

- Tạo tài khoản Stripe, lấy **Test secret key** (vd `sk_test_...`).
- Bật Test Mode trên dashboard.

Trong `.env`:

```
STRIPE_SECRET_KEY=sk_test_...
STRIPE_SUCCESS_URL=http://localhost:3000/payment/stripe/success
STRIPE_CANCEL_URL=http://localhost:3000/payment/stripe/cancel
```

4.2 PayPal Sandbox

Hướng dẫn:

- Vào developer.paypal.com, đăng nhập bằng PayPal thật.
- Tạo:
 - **Business (merchant)** sandbox account.
 - **Personal (buyer)** sandbox account.developers.
- Lấy client ID + secret cho REST API (PayPal Checkout).

Trong `.env`:

```
PAYPAL_CLIENT_ID=...
PAYPAL_CLIENT_SECRET=...
PAYPAL_RETURN_URL=http://localhost:3000/payment/paypal/success
PAYPAL_CANCEL_URL=http://localhost:3000/payment/paypal/cancel
```

4.3 VNPay Sandbox

Hướng dẫn chung (lab có thể dùng sandbox public info):

- Sử dụng VNPay sandbox endpoints:
 - <https://sandbox.vnpayment.vn/paymentv2/vpcpay.html>
- Dùng demo TmnCode (vd `DEMOV210`) và demo secret key như tài liệu hoặc repo mẫu.

Trong `.env`:

```
VNPAY_TMN_CODE=DEMOV210
VNPAY_HASH_SECRET=your_demo_secret
VNPAY_RETURN_URL=http://localhost:3000/payment/vnpay/success
VNPAY_BASE_URL=https://sandbox.vnpayment.vn/paymentv2/vpcpay.html
```

5. Backend – Payment Router Design

File: `backend/routers/payments.py`

5.1 Shared Payment Model (Optional)

SV có thể reuse `PaymentDB` từ Session 13 mock hoặc tạo mới:

Fields:

- `id`, `order_id`, `provider` (`stripe`, `paypal`, `vnpay`).
- `amount`, `currency`.
- `status` (`PENDING`, `SUCCEEDED`, `CANCELED`, etc.).
- `provider_session_id` (Stripe session id, PayPal order id, VNPay transaction ref).

Yêu cầu lab (tùy mức độ):

- **Basic**: chỉ cần `order.status` cập nhật, không cần bảng `PaymentDB`.
- **Advanced**: tạo `PaymentDB` để lưu thông tin từng lần thanh toán.

5.2 Stripe: Create Checkout Session Endpoint

Endpoint: `POST /payments/stripe/create-session`

Requirement:

- Input body: `{ "order_id": number }`.
- Check:
 - Order tồn tại.
 - Order chưa được thanh toán (`status != PAID`).
- Use Stripe SDK + secret key để tạo Checkout Session.

Expected behavior:

- Lấy line items từ order (items đã lưu trong database).
- Set currency, amount theo order.
- Set `success_url` & `cancel_url` từ config.
- Trả về `{ "url": "https://checkout.stripe.com/..." }`.

Sinh viên code (pseudo):

```
import stripe
from fastapi import APIRouter, Depends, HTTPException
```

```

from sqlalchemy.orm import Session

stripe.api_key = settings.STRIPE_SECRET_KEY

@router.post("/stripe/create-session")
def create_stripe_session(order_id: int, db: Session =
Depends(get_db), user=Depends(get_current_user)):
    # validate order & user
    # build line_items from Order.items
    session = stripe.checkout.Session.create(
        mode="payment",
        line_items=[...],
        success_url=settings.STRIPE_SUCCESS_URL +
f"?order_id={order.id}",
        cancel_url=settings.STRIPE_CANCEL_URL +
f"?order_id={order.id}",
    )
    return {"url": session.url}

```

5.3 PayPal: Create Order Endpoint

Endpoint: `POST /payments/paypal/create-order`

Requirements:

- Body: `{ "order_id": number }`.
- Use PayPal REST API + client ID/secret để tạo một "PayPal order".

Basic flow:

1. Backend gửi request tới PayPal API (`v2/checkout/orders`):
 - Amount = `order.total_amount`.
2. PayPal trả về order id + approve link.
3. Backend trả về `{ "approve_url": "https://...paypal.com/checkoutnow?token=..." }`.

Sinh viên implement:

- Gọi PayPal API bằng `httpx` hoặc `requests`.
- Lấy `links` từ response, tìm link có `rel == "approve"`.

5.4 VNPay: Create Payment URL Endpoint

Endpoint: `POST /payments/vnpay/create-url`

Requirements:

- Body: `{ "order_id": number }`.
- Build query parameters theo VNPay docs:
 - `vnp_Amount` (`total_amount * 100`).
 - `vnp_OrderInfo`, `vnp_TxnRef` (unique ID).
 - `vnp_ReturnUrl`, `vnp_Url` sandbox.
- Generate HMAC signature using `VNPAY_HASH_SECRET`.
- Redirect URL:
`https://sandbox.vnpayment.vn/paymentv2/vpcpay.html + query`.

Backend trả về `{ "url":`

`"https://sandbox.vnpayment.vn/paymentv2/vpcpay.html?..."`
`}`.

5.5 Success & Cancel Handling (Basic Version)

Phiên bản đơn giản (ít webhook):

- Khi user được redirect về `success` URL (frontend), query string chứa info:
 - Stripe: `?order_id=...` (bạn set lúc tạo session).
 - PayPal: `?token=..., &PayerID=...` (sau khi approve).`developers.xsolla+1`
 - VNPay: `?vnp_ResponseCode=00&vnp_TxnRef=...` v.v

Backend requirement (simplified):

- Endpoint `POST /payments/confirm`
 - Body: `{ "order_id": number, "provider": "stripe" | "paypal" | "vnpay" }`.
 - Backend update `order.status` → `PAID`.

Như vậy:

- LAB không bắt sinh viên xử lý webhook full, chỉ cần flow end-to-end với callback + manual confirm.

6. Frontend – Payment Page Design

6.1 Route & Components

New route:

- `/orders/:id/payment`

Component: `OrderPaymentPage`

Đề bài yêu cầu:

- Lấy order thông qua `ordersApi.getOrderById(id)`.
- Nếu order `status === "PAID"` → hiển thị “Order already paid”.
- Cho phép user chọn phương thức thanh toán:
 - Radio buttons: Stripe, PayPal, VNPay.
- Nút “Pay Now”:
 - Gọi endpoint tương ứng.
 - Redirect user đến cổng thanh toán (`window.location.href = url`).

6.2 Stripe Payment Flow (Frontend)

Tasks:

- Khi chọn Stripe và click “Pay Now”:
 - Call `POST /payments/stripe/create-session` với `order_id`.
 - Get `{ url }` from response.
 - `window.location.href = url`.

Success page (`/payment/stripe/success`):

- Read `order_id` from query.
- Call `POST /payments/confirm` with `{ order_id, provider: "stripe" }`.

- Show message “Stripe payment successful”.
- Link back to `/orders/:id` to view order status.

Cancel page (`/payment/stripe/cancel`):

- Show “Payment canceled” and link back to `/orders/:id/payment`.

6.3 PayPal Payment Flow (Frontend)

Tasks:

- Khi chọn PayPal và click “Pay Now”:
 - Call `POST /payments/paypal/create-order`.
 - Get `{ approve_url }`.
 - `window.location.href = approve_url`.

Success page (`/payment/paypal/success`):

- Read `token` (PayPal order id) and optional `order_id` from query.
- (Simplified) Call `POST /payments/confirm` with `{ order_id, provider: "paypal" }`.
- Show message “PayPal payment successful”.

Cancel page (`/payment/paypal/cancel`):

- Show “PayPal payment canceled” and link back to payment page.

(LAB nâng cao: SV có thể gọi thêm PayPal capture API trước khi confirm.)

6.4 VNPay Payment Flow (Frontend)

Tasks:

- Khi chọn VNPay và click “Pay Now”:
 - Call `POST /payments/vnpay/create-url`.
 - Get `{ url }`.
 - Redirect to VNPay sandbox page.github+1

Success page (`/payment/vnpay/success`):

- Read `vnp_ResponseCode` and `vnp_TxnRef` from query.

- If `vnp_ResponseCode === "00"` → call `POST /payments/confirm` with `{ order_id, provider: "vnpay" }`.
- Show message "VNPay payment successful".

Cancel / error case:

- If `code !== "00"`, show failure and do not confirm.

7. Lab Tasks – Step-by-Step (Student Version)

Bạn có thể giao lab theo checklist rõ ràng:

Task 1 – Prepare Sandbox Accounts

1. Create Stripe account and enable Test Mode; add `STRIPE_SECRET_KEY` to `.env`.
2. Create PayPal Developer account, sandbox Business & Personal accounts, get client ID/secret.developers.
3. Configure VNPay sandbox settings (TmCode, secret, return URL).

Deliverable: `.env.example` file with placeholders.

Task 2 – Backend: Payment Router

1. Create `payments.py` router with endpoints:
 - `POST /payments/stripe/create-session`
 - `POST /payments/paypal/create-order`
 - `POST /payments/vnpay/create-url`
 - `POST /payments/confirm`
2. Integrate with `OrderDB`:
 - Validate order exists and belongs to current user.
 - Make sure only orders with `status !== PAID` can be paid.
3. Implement simple confirm logic:
 - `confirm` sets `order.status = "PAID"` and returns updated order.

Deliverable:

- Screenshot / code snippet of each endpoint.

Task 3 – Frontend: Payment Page

1. Add route `/orders/:id/payment`.
2. Implement `OrderPaymentPage`:
 - Fetch order by id.
 - Show order info (id, total, status).
 - If status is `PAID`, show “Order already paid” and disable payment.
 - Let user choose payment method (Stripe / PayPal / VNPay).
 - “Pay Now” button:
 - Calls the correct endpoint.
 - Redirects to provider URL.

Deliverable:

- Screenshot of Payment Page with 3 methods.

Task 4 – Frontend: Success / Cancel Pages

1. Add routes:
 - `/payment/stripe/success`, `/payment/stripe/cancel`
 - `/payment/paypal/success`, `/payment/paypal/cancel`
 - `/payment/vnpay/success`, `/payment/vnpay/cancel`
2. For each success page:
 - Read necessary query parameters (`order_id`, `token`, etc.).
 - Call `POST /payments/confirm`.
 - Display clear status and link back to `OrderDetailPage`.
3. For cancel pages:
 - Show proper message and link back to Payment Page.

Deliverable:

- Screenshots of success & cancel pages for each provider.

Task 5 – End-to-End Test Scenarios

Students should test:

- Scenario 1 (Stripe):
 - Create order from cart.

- Pay with Stripe Test Mode using card 4242 4242 4242 4242.stripe+4
 - After success, check that `order.status = "PAID"`.
- Scenario 2 (PayPal):
 - Pay with PayPal sandbox Personal account.
 - After success, confirm order and see status changed to PAID.
- Scenario 3 (VNPAY):
 - Pay via VNPAY sandbox URL, get response code 00.
 - Confirm order and verify PAID status.

Deliverables:

- Brief test report describing each scenario and including screenshots.

8. Optional Advanced Tasks (Bonus)

Làm phần này cho SV khá / có thêm buổi:

- Implement **webhooks**:
 - Stripe webhook endpoint to update order when payment event completes.
 - PayPal webhook to receive payment notifications.
- Store payment details in **PaymentDB**:
 - Keep provider session id, status, and timestamps for auditing.
- Add “Paid via X” label in **OrderDetailPage**:
 - Example: “Payment provider: Stripe (Test Mode)”.

Điểm bonus nếu:

- Flow xử lý failure / cancel rõ ràng.
- Không thể “double charge” một order (idempotent confirm).