# TẠO THANH TOÁN BẰNG API

Tác giả: PA

### 1. Xem Video Hướng Dẫn

https://www.youtube.com/watch?v=GEnt7jmIUiI

### 2. Đăng Ký Merchant Môi Trường Test

- Tên Website và URL có thể đặt tùy ý vì không quan trọng. Sẽ được gửi mail thông báo kích hoạt và mail hướng dẫn các thông tin cần thiết



### ĐĂNG KÝ MERCHANT MÔI TRƯỜNG TEST

<b>Q</b>	Ecommerce Website
<b>\$</b>	www.magicshop.com.vn
	oephuocan@gmail.com
0-	Mật khẩu
0-	Nhập lại mật khẩu
Nhập mã xác nhận	
	9VVAYT 2
	Đăng ký



### 3. Tạo thư mục Helpers và tạo lớp VnPayLibrary

- Copy nội dung của VnPayLibrary trong file sau : <a href="https://docs.google.com/document/d/1XaOZJRJCyt0\_WcHsxXi1xrzcC\_8iKlUzAxhk\_V40j8/edit">https://docs.google.com/document/d/1XaOZJRJCyt0\_WcHsxXi1xrzcC\_8iKlUzAxhk\_V40j8/edit</a>

### 4. Tạo lớp VnPayResponseModel

- Copy nội dung của VnPayResponseModel cũng trong file trên

Và thêm phần dưới đây vào

```
public class VnPaymentRequestModel
{
      public int OrderId { get; set; }
      public double Amount { get; set; }
      public DateTime CreatedDate { get; set; }
}
```

## 5. Định nghĩa các hằng số trong appsettings.json

```
"VnPay": {
  "TmnCode": "HS7H0S37",
  "HashSecret": "SRXWHXLMCBGSRRMNOCABIXMODZSMHGXH",
  "BaseUrl": "https://sandbox.vnpayment.vn/paymentv2/vpcpay.html",
  "Version": "2.1.0",
  "Command": "pay",
  "CurrCode": "VND",
  "Locale": "vn",
  "PaymentBackReturnUrl": "https://localhost:44322/Cart/PaymentCallBack"
},
```

- Các thông tin về TmnCode, HashSecret và BaseUrl thì dựa theo mail được gửi khi đăng ký Merchant Test

#### 6. Tạo Interface IvnPayService

```
public interface IVnPayService
      string CreatePaymentUrl(HttpContext context, VnPaymentRequestModel model);
      VnPaymentResponseModel PaymentExecute(IQueryCollection collections);
7. Tao lớp VnPayService
public class VnPayService : IVnPayService
      private readonly IConfiguration _config;
      public VnPayService(IConfiguration config)
             _config = config;
      }
      public string CreatePaymentUrl(HttpContext context, VnPaymentReguestModel
model)
             var tick = DateTime.Now.Ticks.ToString();
             var vnpay = new VnPayLibrary();
             vnpay.AddRequestData("vnp_Version", _config["VnPay:Version"]);
             vnpay.AddRequestData("vnp_Command", _config["VnPay:Command"]);
             vnpay.AddRequestData("vnp_TmnCode", _config["VnPay:TmnCode"]);
vnpay.AddRequestData("vnp_Amount", (model.Amount * 100).ToString());
             vnpay.AddRequestData("vnp_CreateDate",
model.CreatedDate.ToString("yyyyMMddHHmmss"));
             vnpay.AddRequestData("vnp_CurrCode", _config["VnPay:CurrCode"]);
             vnpay.AddRequestData("vnp_IpAddr", Utils.GetIpAddress(context));
             vnpay.AddRequestData("vnp_Locale", _config["VnPay:Locale"]);
             vnpay.AddRequestData("vnp_OrderInfo", "Thanh toán đơn hàng:" +
model.OrderId);
             vnpay.AddRequestData("vnp_OrderType", "other");
             vnpay.AddRequestData("vnp_ReturnUrl",
_config["VnPay:PaymentBackReturnUrl"]);
             vnpay.AddRequestData("vnp_TxnRef", tick);
             var paymentUrl = vnpay.CreateRequestUrl(_config["VnPay:BaseUrl"],
_config["VnPay:HashSecret"]);
             return paymentUrl;
      }
      public VnPaymentResponseModel PaymentExecute(IQueryCollection collections)
             var vnpay = new VnPayLibrary();
             foreach (var (key, value) in collections)
```

```
{
                   if (!string.IsNullOrEmpty(key) && key.StartsWith("vnp_"))
                          vnpay.AddResponseData(key, value.ToString());
                   }
             }
             var vnp_orderId = Convert.ToInt64(vnpay.GetResponseData("vnp_TxnRef"));
             var vnp_TransactionId =
Convert.ToInt64(vnpay.GetResponseData("vnp_TransactionNo"));
             var vnp_SecureHash = collections.FirstOrDefault(p => p.Key ==
"vnp_SecureHash").Value;
             var vnp_ResponseCode = vnpay.GetResponseData("vnp_ResponseCode");
             var vnp_OrderInfo = vnpay.GetResponseData("vnp_OrderInfo");
             bool checkSignature = vnpay.ValidateSignature(vnp_SecureHash,
_config["VnPay:HashSecret"]);
             if (!checkSignature)
                   return new VnPaymentResponseModel
                          Success = false
                   };
          return new VnPaymentResponseModel {
                  Success = true,
                  PaymentMethod = "VnPay",
                    OrderDescription = vnp_OrderInfo,
                    OrderId = vnp_orderId.ToString(),
                  TransactionId = vnp_TransactionId.ToString(),
                  Token = vnp_SecureHash,
                  VnPayResponseCode = vnp_ResponseCode,
             };
      }
```

8. Chỉnh sửa view Checkout để điều hướng tới VnPay (Đã hoàn thiện trước đó )

#### 9. Chỉnh sửa CartController

### a. Chỉnh Phương Thức ProcessCheckOutVnPay

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> ProcessCheckoutVNPay()
{
    var user = User.FindFirst(ClaimTypes.NameIdentifier);
    if (user == null)
    {
        return RedirectToAction("Login", "Account");
    }
    string userId = user.Value;
```

```
var listCart = await _DbContext.CartItems
                    .Include(ci => ci.Product)
                    .Where(ci => ci.UserId == userId)
                   .ToListAsync();
      double total = 0;
      foreach (var cartItem in listCart)
             total += cartItem.Product.Price * cartItem.Quantity;
      var vnPayModel = new VnPaymentRequestModel
    {
        Amount = total,
        CreatedDate = DateTime.Now,
        OrderId = 1,
    };
      return Redirect(_vnPayService.CreatePaymentUrl(HttpContext, vnPayModel));
}
      b. Thêm Phương Thức PaymentCallBack
public async Task<IActionResult> PaymentCallBack()
    var response = _vnPayService.PaymentExecute(Request.Query);
    if (response == null || response.VnPayResponseCode != "00")
        return RedirectToAction("ShoppingCart");
      try
             var user = User.FindFirst(ClaimTypes.NameIdentifier);
             if (user == null)
             {
                   return RedirectToAction("Login", "Account");
             string userId = user.Value;
             var listCart = await _DbContext.CartItems
                    .Include(ci => ci.Product)
                    .Where(ci => ci.UserId == userId)
                   .ToListAsync();
             double total = 0;
             foreach (var cartItem in listCart)
                   total += cartItem.Product.Price * cartItem.Quantity;
             var order = new Order
                   UserId = userId,
                   Status = DeliveryStatus.Processing.ToString(),
                   Date = DateTime.Now,
                   ShippingCost = 0,
                   Total = total,
            Paid = true,
                   PayMethod = PayMethod. VNPay. ToString(),
                   OrderDetails = new List<OrderDetail>()
```

```
};
            await _DbContext.Orders.AddAsync(order);
            await _DbContext.SaveChangesAsync();
            foreach (var cartItem in listCart)
                   var orderDetail = new OrderDetail
                         OrderId = order.Id,
                         ProductId = cartItem.ProductId,
                         Quantity = cartItem.Quantity,
                         ProductCost = cartItem.Product.Price * cartItem.Quantity,
                         UnitPrice = cartItem.Product.Price
                   await _DbContext.OrderDetails.AddAsync(orderDetail);
            }
            _DbContext.CartItems.RemoveRange(listCart);
            await _DbContext.SaveChangesAsync();
            // Gửi email xác nhận đặt hàng
            var email = User.FindFirstValue(ClaimTypes.Email);
            var subject = "MAGIC SHOP - Xác nhận đặt hàng ( phương thức thanh toán
COD)";
            var htmlMessage =
                   "<p>Xin chào!,<math></p>\r\n
                   "Cảm ơn bạn đã đặt hàng tại cửa hàng của chúng tôi. Đơn hàng
của bạn đã được nhận và đang được xử lý." +
                   "Chúng tôi hy vọng bạn sẽ hài lòng về sản phấm." +
                   "Chúc bạn một ngày tốt lành♥□" +
                   "<br>" +
                   "Thân mến," +
                   "Magic Shop";
            await _sendMailService.SendEmailAsync(email, subject, htmlMessage);
            return RedirectToAction("Profile", "User");
      }
      catch
      {
            return RedirectToAction("ShoppingCart");
      }
}
```

### 10. Kiểm thử kết quả

Dùng các api thông tin thẻ được cung cấp ở trang sau : <a href="https://sandbox.vnpayment.vn/apis/vnpay-demo/">https://sandbox.vnpayment.vn/apis/vnpay-demo/</a>

→ Đây là thông tin dùng để thực hiện thanh toán thành công với VnPay (Copy qua để test )

Số Thẻ: 9704198526191432198

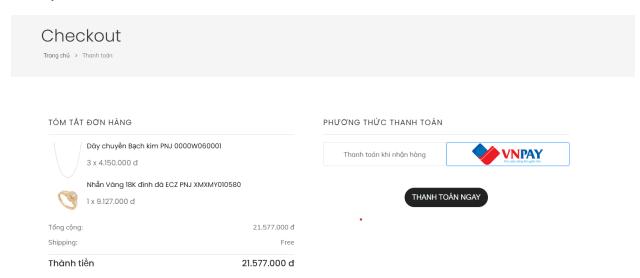
Tên Chủ Thẻ: NGUYEN VAN A

Ngày Phát Hành: 07/15

OTP: 123456

#### **DEMO**

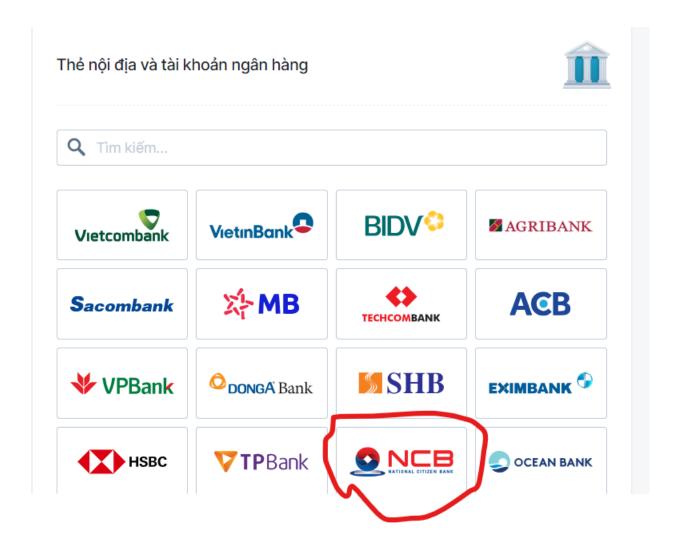
+ Chọn thanh toán VNPAY



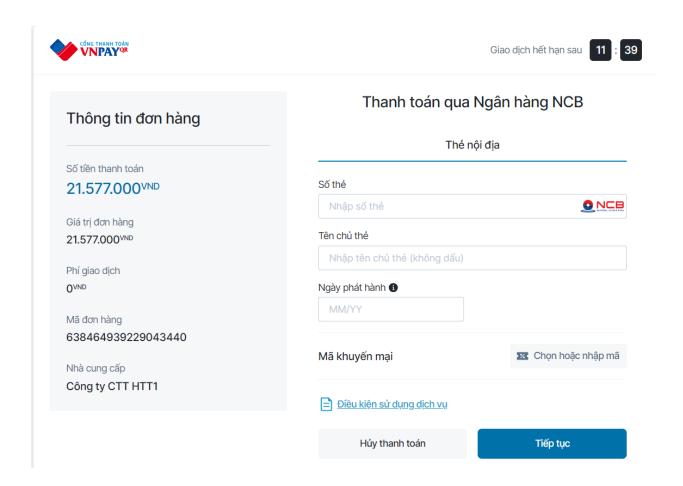
+ Chọn thanh toán bằng thẻ nội địa và tài khoản ngân hàng



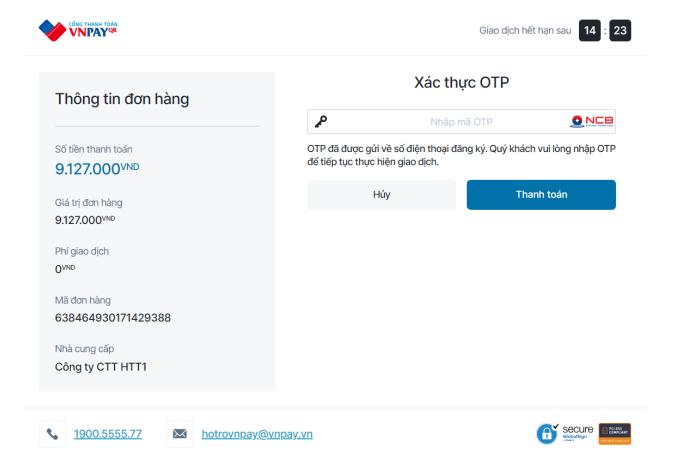
# + Chọn ngân hàng **NCB**



+ Nhập các thông tin được cung cấp ở trên hoặc theo đường link nếu muốn test các trường hợp



### + Nhập mã OTP



### + Đơn hàng được lưu và thanh toán thành công

