

Git TRIỂN KHAI PHÁT TRIỂN SẢN PHẨM

TIỀN TRÌNH 1: KHỞI TẠO VÀ XÂY DỰNG BAN ĐẦU

Bước 1: Tạo và làm việc trên nhánh riêng (nếu chưa có)

Đặt tên theo quy tắc: feature/ten-thanh-vien-ten-tinh-nang.

Ví dụ:

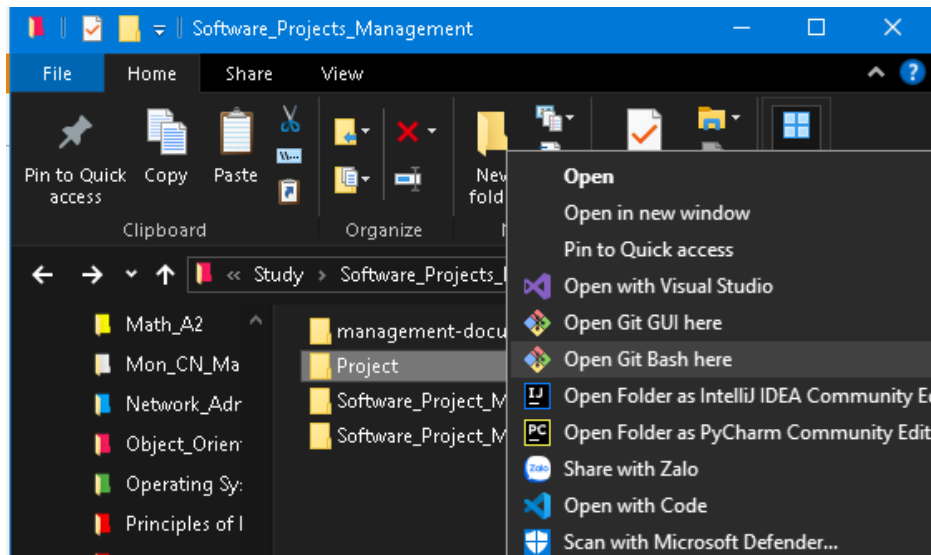
feature/pchai-orderstatus...

feature/nphuy-brand...

feature/nvkha-categories...

feature/bnmhuy-posttype...

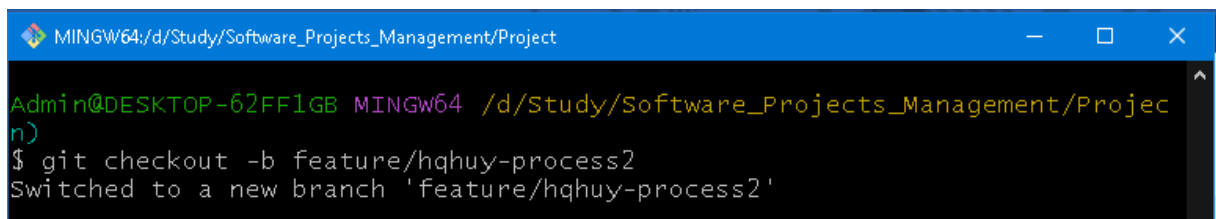
- Trong thư mục chứa thư mục dự án >chuột trái> Open Git Bash here



Bước 2: Chạy lệnh sau:

Tạo một nhánh mới và chuyển sang nhánh đó

git checkout -b feature/tinh-nang-cua-ban



Bước 3: Kéo cập nhật mới nhất từ các thành viên khác từ nhánh main

Lưu trữ các thay đổi hiện tại (nếu có)

git stash

Chuyển về nhánh chính

git checkout main

Kéo mã nguồn mới nhất từ remote

git pull origin main

Quay lại nhánh của bạn

git checkout feature/tinh-nang-cua-ban

```
Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (feature/hqhuy-process2)
$ git stash
Saved working directory and index state WIP on feature/hqhuy-process2: 75c b13c Merge pull request #19 from hkhuang07/feature/nvkha-categories-topics

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (feature/hqhuy-process2)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.
```

```
Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (main)
$ git pull origin main
From https://github.com/hkhuang07/NewsMart-ECommerce-Laravel
* branch      main      -> FETCH_HEAD
Already up to date.
```

```
Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (main)
$ git checkout feature/hqhuy-process2
Switched to branch 'feature/hqhuy-process2'

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (feature/hqhuy-process2)
$
```

Bước 4: Tiến hành thực hiện các tác vụ của từng thành viên:

...

Sau khi clone repository về mọi người copy folder *NewsMart_ECommerce_Laravel* vào www của Wamp64

Sau khi cập nhật những file mới mọi người replace các file mới này với file cùng tên trong thư mục có chứa .gits

...

Lộ Trình Phát Triển Tiếp Theo: Xây dựng Logic Cốt Lõi và Tương Tác

Các module tiếp theo cần tập trung vào logic giao dịch, xử lý đa vai trò (multi-role) và tương tác người dùng (UX).

Giai đoạn 1: Xử lý Xác thực Nâng cao & Nội dung Cơ bản (Module 1, 5, 10)

STT	Module	Chức năng cần xây dựng	Hướng dẫn triển khai (Chi tiết)
1.	Module Người dùng & Phân quyền	Duyệt/Từ chối Đối tác	1. Route/View: Tạo trang quản lý PartnershipRequests (hoặc view Admin để hiển thị Users có RoleID = Pending_Saler và Pending_Shipper). 2. Controller/Logic: Xây dựng action approvePartner(\$user_id). Logic sẽ cập nhật trường role_id của user thành Saler hoặc Shipper tương ứng.
2.	Module Người dùng & Phân quyền	Theo dõi Lịch sử Hoạt động (UserActivities)	1. Model: Tạo Model UserActivity với các trường: user_id, action (ví dụ: 'PRODUCT_CREATE', 'ORDER_STATUS_UPDATE'), details (JSON), ip_address. 2. Implement Logic: Sử dụng Observer hoặc Events/Listeners của Laravel. Khi Admin, Manager, hoặc Saler thực hiện các hành động quan trọng (như CRUD sản phẩm, duyệt bài viết, thay đổi trạng thái đơn hàng), tự động gọi hàm để tạo bản ghi mới trong bảng UserActivities.
3.	Module Bài viết (Posts)	CRUD Bài viết (User/Saler)	1. Model: Tạo Model Post (liên kết với User, Topic, PostType). Thêm cột status_id (mặc định là 'Pending'). 2. Controller/View: Xây dựng giao diện cho User/Saler để tạo và quản lý bài viết của họ. Dùng middleware để đảm bảo họ chỉ có thể sửa/xóa bài viết có user_id khớp với ID của họ.
4.	Module Tìm kiếm & Hiển thị	Hiển thị Danh sách Sản phẩm/Bài viết	1. Controller: Xây dựng các hàm index() cho ProductsController và PostsController. 2. Logic Sắp xếp: Ban đầu, sắp xếp theo created_at mới nhất. Sau này, thêm logic orderBy('views') hoặc orderBy('average_rate').

Giai đoạn 2: Xây dựng Logic Mua sắm (Module 4, 6, 7)

Giai đoạn này khó , tập trung vào các giao dịch.

STT	Module	Chức năng cần xây dựng	Hướng dẫn triển khai (Chi tiết)
5.	Module Sản phẩm (Products)	Quản lý Sản phẩm (Admin/Manager)	1. Controller: Triển khai các hàm CRUD cho Products và ProductImages. 2. Tồn kho: Thêm cột stock_quantity vào bảng Products.
6.	Module Giỏ hàng (Carts)	Quản lý Giỏ hàng (CRUD Carts)	1. Model: Tạo Model Cart (liên kết User, Product, Quantity). 2. API/Controller: Xây dựng API (hoặc Controller) cho các thao tác: addToCart(\$product_id, \$qty), updateCart(\$cart_item_id, \$qty), removeCart(\$cart_item_id).
7.	Module Tương tác & Đánh giá	Yêu thích Sản phẩm (ProductFavorites)	1. Model: Tạo Model ProductFavorite (liên kết User, Product). 2. Controller/API: Cho phép User thêm/xóa sản phẩm khỏi yêu thích.
8.	Module Tương tác & Đánh giá	Tương tác Bài viết & Bình luận	1. Model: Tạo PostInteraction (Likes/Shares) và Comment. 2. API/Controller: Xây dựng logic ghi nhận PostInteraction và hiển thị/lưu trữ Comments.
9.	Module Giỏ hàng & Đơn hàng	Đặt hàng (Checkout)	1. Controller (Checkout): Chức năng này lấy dữ liệu từ Carts của User. 2. Logic: Tạo bản ghi mới trong Orders, chuyển dữ liệu sản phẩm sang OrderItems (ghi lại PriceAtOrder), và tạo ShippingInformation. 3. Xóa Giỏ hàng: Xóa các mục trong bảng Carts sau khi tạo đơn hàng.

Giai đoạn 3: Xây dựng Logic Đa Vai trò & Báo cáo (Module 8, 9, 12)

STT	Module	Chức năng cần xây dựng	Hướng dẫn triển khai (Chi tiết)
10.	Module Saler	Quản lý Sản phẩm theo Saler	1. Model: Thêm cột saler_id vào bảng Products (liên kết với Users). 2. Logic: Khi Saler truy cập CRUD sản phẩm, áp dụng điều kiện WHERE products.saler_id = Auth::id().
11.	Module Saler & Shipper	Cập nhật Trạng thái Đơn hàng	1. Controller (Order Management): Viết logic để chỉ cho phép Saler cập nhật các trạng thái nhất định (ví dụ: 'Confirmed', 'Preparing'), và Shipper cập nhật các trạng thái liên quan đến giao hàng ('Picking Up', 'Delivering', 'Delivered').
12.	Module Shipper	Gán Đơn hàng & Theo dõi	1. Model: Tạo Model ShipperAssignment (liên kết Order, User/Shipper). 2. Logic: Shipper xem danh sách Orders chưa có Shipper. Khi họ "Pick Up", tạo bản ghi mới trong ShipperAssignment.
13.	Module Thống kê	Báo cáo Doanh thu (Saler/Admin)	1. Model: Đảm bảo OrderTransactions ghi lại Amount. 2. Controller: Viết truy vấn GROUP BY (SQL) trên bảng OrderTransactions và OrderItems để tính tổng doanh thu và lợi nhuận, lọc theo saler_id và nhóm theo thời gian (DATE()).
14.	Module Thông báo	Hệ thống Notifications	1. Model: Tạo Model Notification (user_id, message, type, is_read). 2. Implement Logic: Sử dụng Events & Listeners của Laravel. Khi OrderStatus thay đổi, StockQuantity thấp, hoặc bài viết được Approved, kích hoạt Listener để tạo bản ghi mới trong bảng Notifications.

Yêu cầu về chức năng (Functional Requirements)

- Quản trị (Admin Panel)

+ Quản lý người dùng và phân quyền

☐ Theo dõi Lịch sử Hoạt động: Ghi nhận mọi thao tác quan trọng của Admin/Manager/Saler vào bảng UserActivities.

☐ Duyệt/Từ chối Đối tác: Cập nhật trường RoleID của người dùng trong bảng Users để xác nhận quyền Saler hoặc Shipper.

- ☐ Quản lý Phân quyền: Thêm, sửa, xóa các vai trò trong bảng `Roles` (Admin, Product Manager, Content Editor, Saler, Driver).
- ☐ Toàn quyền thao tác CRUD (Create, Read, Update, Delete) trên bảng User.
- + Quản lý hệ thống
- ☐ Quản lý Danh mục & Thương hiệu: Thao tác CRUD trên các bảng Categories và Brands.
- ☐ Quản lý Chủ đề : Thao tác CRUD trên bảng Topics
- ☐ Quản lý các thành phần khác trong hệ thống: Thao tác CRUD trên các bảng Roles, PostTypes, PostStatus, OrderStatuses, Reviews, Configurations .- Quản lý (Admin và Manager)
- + Quản lý sản phẩm và bài viết
- ☐ Kiểm duyệt Bài viết: Thiết lập trạng thái ("Pending", "Approved", "Rejected") trên các Posts mới được gửi bởi User/Saler.
- ☐ Quản lý Toàn diện Sản phẩm: Thao tác CRUD trên Products, ProductImages.
- + Quản lý đơn hàng
- ☐ Cập nhật trạng thái đơn hàng OrderStatus (từ Đặt hàng, Chờ xác nhận, Đang giao, Đã giao, Hủy) và phân loại cho từng đối tác.
- ☐ Xem Giao dịch: Tra cứu chi tiết thanh toán trong bảng OrderTransactions.
- ☐ Thao tác CRUD đơn hàng trong trường hợp cần thiết.
- Đối tác bán hàng (Saler)
- + Quản lý sản phẩm
- ☐ Thêm/Sửa/Xóa Sản phẩm: Saler chỉ có quyền quản lý các Products mà họ tạo (cần thêm cột SalerID vào bảng Products hoặc áp dụng logic phân quyền).
- ☐ Cập nhật Tồn kho Tự động: Trường StockQuantity trong Products tự động trừ đi số lượng trong OrderItems sau khi đơn hàng được xác nhận thanh toán/giao dịch.
- ☐ Cảnh báo Tồn kho: Hệ thống kiểm tra StockQuantity so với ngưỡng cảnh báo trong Configurations và gửi thông báo (Notifications) đến Saler.

+ Quản lý đơn hàng

☐ Tiếp nhận Đơn hàng: Tự động hiển thị các Orders có SallerID khớp với ID của Saller.

☐ Cập nhật Trạng thái: Saller cập nhật OrderStatusID (ví dụ: "Đã xác nhận", "Đã chuẩn bị hàng") sau khi nhận thông tin từ khách hoặc Shipper..

+ Thống kê doanh thu:

☐ Báo cáo Doanh thu: Cung cấp tổng quan và chi tiết doanh thu dựa trên dữ liệu Amount trong OrderTransactions và PriceAtOrder trong OrderItems, nhóm theo ngày/tháng/năm.

- Đối tác vận chuyển

+ Nhận đơn hàng:

☐ Gán Đơn hàng: Cơ chế cho phép Driver chọn nhận (pick up) đơn hàng chưa được gán, tạo một bản ghi mới trong bảng DriverAssignments.

+ Cập nhật trạng thái:

☐ Cập nhật Trạng thái Giao hàng: Shipper truy cập các đơn hàng đã được gán cho mình (DriverAssignments) và cập nhật OrderStatusID trong bảng Orders theo các mốc thời gian (Đã lấy hàng, Đang giao, Thất bại, Đã giao).

☐ Gửi Thông báo: Tự động gửi thông báo (Notifications) cho User và Saller khi trạng thái đơn hàng thay đổi.

- Người dùng cuối (End-User)

+ Mua sắm và đơn hàng:

☐ Quản lý Giỏ hàng: Thao tác CRUD trên bảng Carts (thêm, xóa, thay đổi Quantity).

☐ Đặt hàng: Chuyển dữ liệu từ Carts sang Orders, OrderItems, và tạo bản ghi ShippingInformation.

☐ Thanh toán: Ghi lại kết quả giao dịch vào bảng OrderTransactions.

+ Tương tác nội dung:

☐ Đánh giá Sản phẩm: Người dùng đăng Reviews (Rating, Content) sau khi mua. Hệ thống tự động cập nhật AverageRate trong Products.

☐ Tương tác Bài viết: Tạo bản ghi trong PostInteractions (cho

Likes/Shares) và Comments (cho bình luận).

+ Đăng bài:

☐ Gửi Bài viết: Người dùng tạo Posts mới. Hệ thống mặc định trạng thái là "Pending" chờ kiểm duyệt.

+ Hoạt động người dùng:

☐ Yêu thích Sản phẩm: Thêm/xóa sản phẩm khỏi danh sách yêu thích cá nhân bằng cách tạo/xóa bản ghi trong ProductFavorites.

☐ Quản lý thông tin người dùng

☐ Đổi mật khẩu, quên mật khẩu.

☐ Nhận Thông báo: Hiện thị thông báo từ bảng Notifications.

Bước 14: Cập nhật thay đổi

- Sau khi đã hoàn thành những đoạn code mới trên repository của bạn hãy thực hiện cập nhật dự án vào repo.

Áp dụng các thay đổi đã lưu trên nhánh cá nhân
git stash pop


```

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (feature/hqhuy-process2)
$ git stash pop
On branch feature/hqhuy-process2
Changes not staged for commit:
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    deleted:    Documents/Process01.docx
    modified:   NewsMart_ECommerce_Laravel/app/Helpers/PermissionHelper
.php
    modified:   NewsMart_ECommerce_Laravel/app/Helpers/helper_functions
.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/CategoryController.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/ConfigurationController.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/ReviewController.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/TopicController.php
    modified:   NewsMart_ECommerce_Laravel/app/Providers/HelperServiceProvider.php
    modified:   NewsMart_ECommerce_Laravel/database/newsmart._sql_server.sql
    modified:   NewsMart_ECommerce_Laravel/database/newsmart_mysql.sql
    modified:   NewsMart_ECommerce_Laravel/database/newsmart_mysql_v1.sql
    modified:   NewsMart_ECommerce_Laravel/database/newsmart_mysql_v2.sql
    modified:   NewsMart_ECommerce_Laravel/lang/vi/validation.php
    modified:   NewsMart_ECommerce_Laravel/package.json
    modified:   NewsMart_ECommerce_Laravel/public/css/custom.css
    modified:   NewsMart_ECommerce_Laravel/public/css/form.css
    modified:   NewsMart_ECommerce_Laravel/public/css/list.css
    modified:   NewsMart_ECommerce_Laravel/resources/views/auth/login.b

```

Thêm các đổi

git add .

#kiểm tra thay đổi

git status

```

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (feature/hqhuy-process2)
$ git add .
warning: in the working copy of 'NewsMart_ECommerce_Laravel/public/css/auth_ver1.css', CRLF will be replaced by LF the next time Git touches it

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Project (feature/hqhuy-process2)
$ git status
On branch feature/hqhuy-process2
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    deleted:    Documents/Process01.docx
    modified:   NewsMart_ECommerce_Laravel/app/Helpers/PermissionHelper.php
    modified:   NewsMart_ECommerce_Laravel/app/Helpers/helper_functions.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/CategoryController.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/ConfigurationController.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/ReviewController.php
    modified:   NewsMart_ECommerce_Laravel/app/Http/Controllers/TopicController.php
    new file:   NewsMart_ECommerce_Laravel/public/css/auth.css
    new file:   NewsMart_ECommerce_Laravel/public/css/auth_ver1.css
    modified:   NewsMart_ECommerce_Laravel/public/css/custom.css
    new file:   NewsMart_ECommerce_Laravel/public/images/favicon.ico
    new file:   NewsMart_ECommerce_Laravel/public/images/favicon1.ico
    new file:   NewsMart_ECommerce_Laravel/public/images/favicon2.ico
    new file:   NewsMart_ECommerce_Laravel/public/images/favicon3.ico
    new file:   NewsMart_ECommerce_Laravel/public/images/favicon4.ico
    new file:   NewsMart_ECommerce_Laravel/public/images/login-banner.jpg

```

chấp nhận các thay đổi
git commit -m "[Message...]"

```

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Projec
t (feature/hqhuy-process2)
$ git commit -m "Update Authentication[Login/Register] UI/UX [10/27/2025]"
[feature/hqhuy-process2 a7b6f43] Update Authentication[Login/Register] UI/
UX [10/27/2025]
45 files changed, 3211 insertions(+), 632 deletions(-)
delete mode 100644 Documents/Process01.docx
create mode 100644 NewsMart_ECommece_Laravel/public/css/auth.css
create mode 100644 NewsMart_ECommece_Laravel/public/css/auth_ver1.css
create mode 100644 NewsMart_ECommece_Laravel/public/images/favicon.ico
create mode 100644 NewsMart_ECommece_Laravel/public/images/favicon1.ico
create mode 100644 NewsMart_ECommece_Laravel/public/images/favicon2.ico
create mode 100644 NewsMart_ECommece_Laravel/public/images/favicon3.ico
create mode 100644 NewsMart_ECommece_Laravel/public/images/favicon4.ico
create mode 100644 NewsMart_ECommece_Laravel/public/images/login-banner.j
pg
create mode 100644 NewsMart_ECommece_Laravel/public/images/login-banner01
.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/login-bg.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/login-bg01.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/login-bg02.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/login-bnr.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/logo.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/logo.png
create mode 100644 NewsMart_ECommece_Laravel/public/images/newsmart_logo.
jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/register-banne
r.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/register-banne
r01.jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/register-bg.jp
g
create mode 100644 NewsMart_ECommece_Laravel/public/images/register-bg01.
jpg
create mode 100644 NewsMart_ECommece_Laravel/public/images/register-bg02.

```

cập nhật lên repository

git push origin feature/ten-nhanh-cua-ban

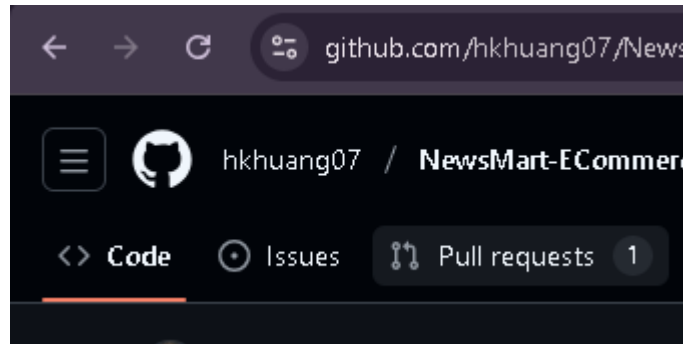
```

Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Projec
t (feature/hqhuy-process2)
$ git push origin feature/hqhuy-process2
Enumerating objects: 83, done.
Counting objects: 100% (83/83), done.
Delta compression using up to 4 threads
Compressing objects: 100% (52/52), done.
Writing objects: 100% (53/53), 377.47 KiB | 4.84 MiB/s, done.
Total 53 (delta 21), reused 2 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (21/21), completed with 18 local objects.
remote:
remote: Create a pull request for 'feature/hqhuy-process2' on GitHub by vi
siting:
remote:   https://github.com/hkhuang07/NewsMart-ECommerce-Laravel/pull/
new/feature/hqhuy-process2
remote:
To https://github.com/hkhuang07/NewsMart-ECommerce-Laravel.git
 * [new branch]      feature/hqhuy-process2 -> feature/hqhuy-process2

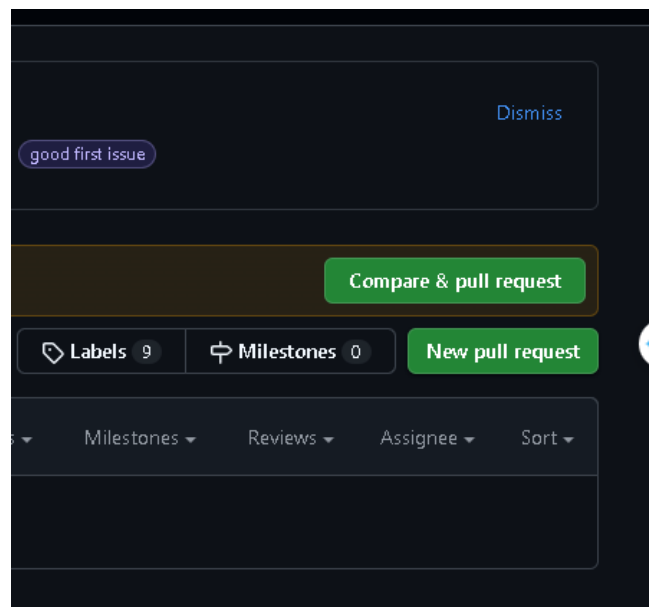
Admin@DESKTOP-62FF1GB MINGW64 /d/Study/Software_Projects_Management/Projec
t (feature/hqhuy-process2)
$

```

- Yêu cầu được gửi lên nhánh main (Compare & Pull Request), sẽ được admin(leader) kiểm duyệt có ảnh hưởng đến code khác hay không để cập nhật tiến độ dự án



- Nếu được **Compare & Pull request** (Dành cho tài khoản tạo repository)



- Tạo **pull request**

- Folder Demo đã được cập nhật vào nhánh main

