*Bài Thực Hành 05 CT299*

**Làm Quen Với GitHub**

**Câu 1:**

**Câu 2:**

1. **Lợi ích của Git**

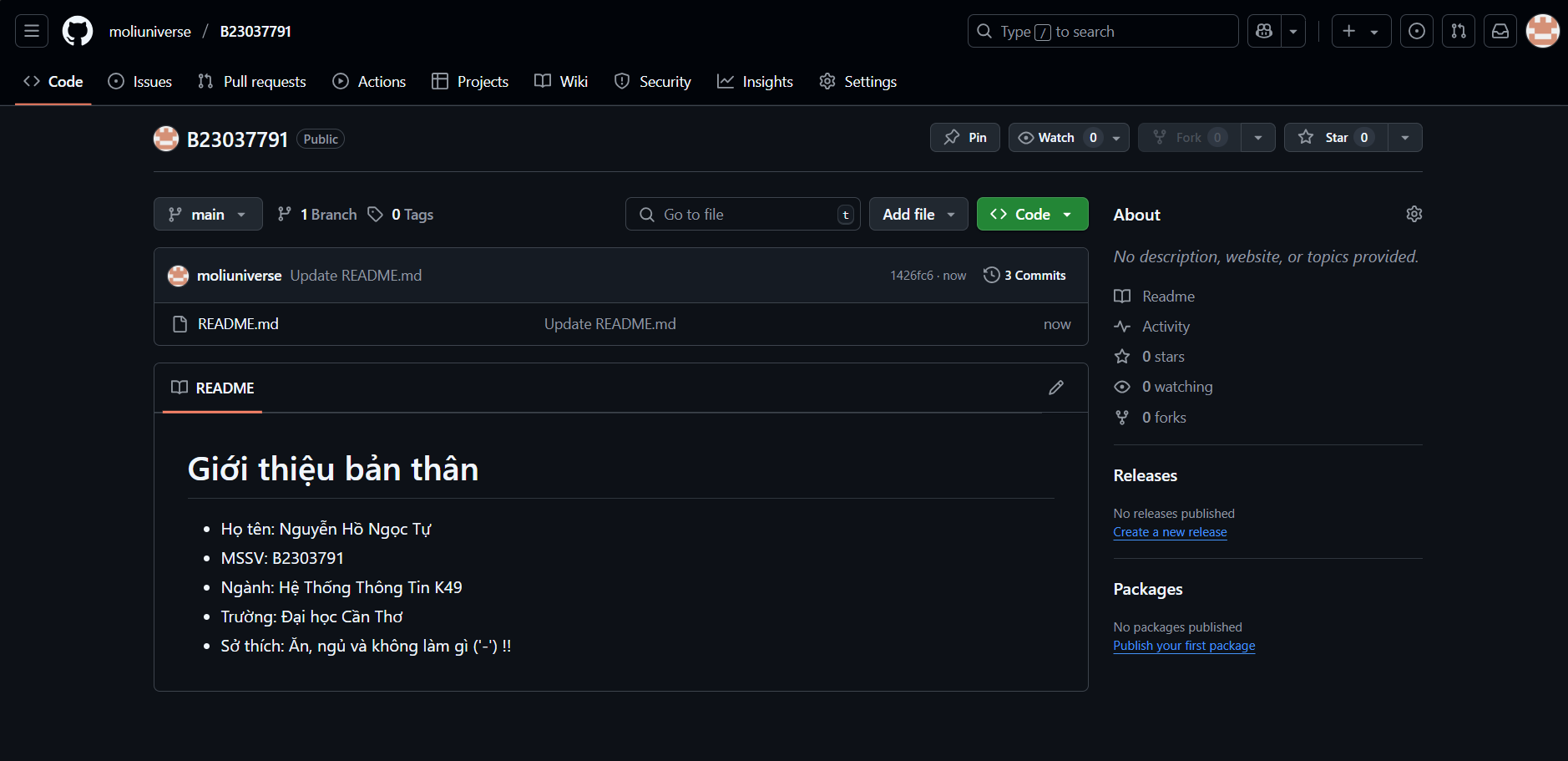
* Lưu lại toàn bộ lịch sử chỉnh sửa của dự án.
* Dễ dàng quay lại phiên bản cũ khi cần.
* Mỗi lập trình viên có bản sao đầy đủ của toàn bộ dự án trên máy cá nhân.
* Có thể làm việc offline và đồng bộ sau.
* Dễ dàng tạo các nhánh để thử nghiệm hoặc phát triển tính năng mới.
* Việc gộp các nhánh rất hiệu quả và an toàn.
* Dữ liệu được lưu trữ ở nhiều nơi.
* Giảm nguy cơ mất mã nguồn khi máy chủ gặp sự cố.
* Các thao tác như commit, diff, merge được thực hiện nhanh chóng vì hoạt động cục bộ.

1. **Lợi ích của GitHub**

* Dễ dàng truy cập và tải mã nguồn từ mọi nơi.
* Có thể chia sẻ với người khác qua đường dẫn.
* Cho phép nhiều người cùng tham gia vào một dự án.
* Theo dõi ai thay đổi gì và khi nào.
* Ghi nhận và theo dõi lỗi, nhiệm vụ và tính năng mới cần làm.
* Hỗ trợ quản lý dự án chuyên nghiệp.
* Thành viên có thể đề xuất thay đổi mã thông qua pull request.
* Quản lý hoặc thành viên khác có thể xem xét trước khi chấp nhận thay đổi.
* Tích hợp các công cụ kiểm thử tự động (GitHub Actions).
* Hỗ trợ tự động hóa triển khai, kiểm thử, build,…
* Là nền tảng phổ biến cho các dự án mã nguồn mở toàn cầu.
* Tạo cơ hội học hỏi, đóng góp và xây dựng cộng đồng.
* Dễ sử dụng, dễ quản lý dự án, commit, lịch sử và các nhánh.

**Câu 3:**

* **Repository** là một "kho lưu trữ" trên GitHub dùng để chứa toàn bộ mã nguồn, tài liệu và lịch sử thay đổi của một dự án phần mềm.
* Nó giúp:
* Quản lý và theo dõi phiên bản của mã nguồn.
* Chia sẻ và cộng tác với người khác.
* Lưu trữ mọi thứ liên quan đến dự án như mã, README, giấy phép (LICENSE), tài liệu,…
* Tổng dung lượng khuyến nghị cho một repository là không quá **1 GB**.
* **Tối đa** 5 GB dung lượng cho mỗi repository miễn phí.
* Nếu vượt quá, bạn sẽ thấy cảnh báo, và nên chuyển sang Git Large File Storage (Git LFS) hoặc phân tách repository.



*Link:* [*moliuniverse/B23037791*](https://github.com/moliuniverse/B23037791)

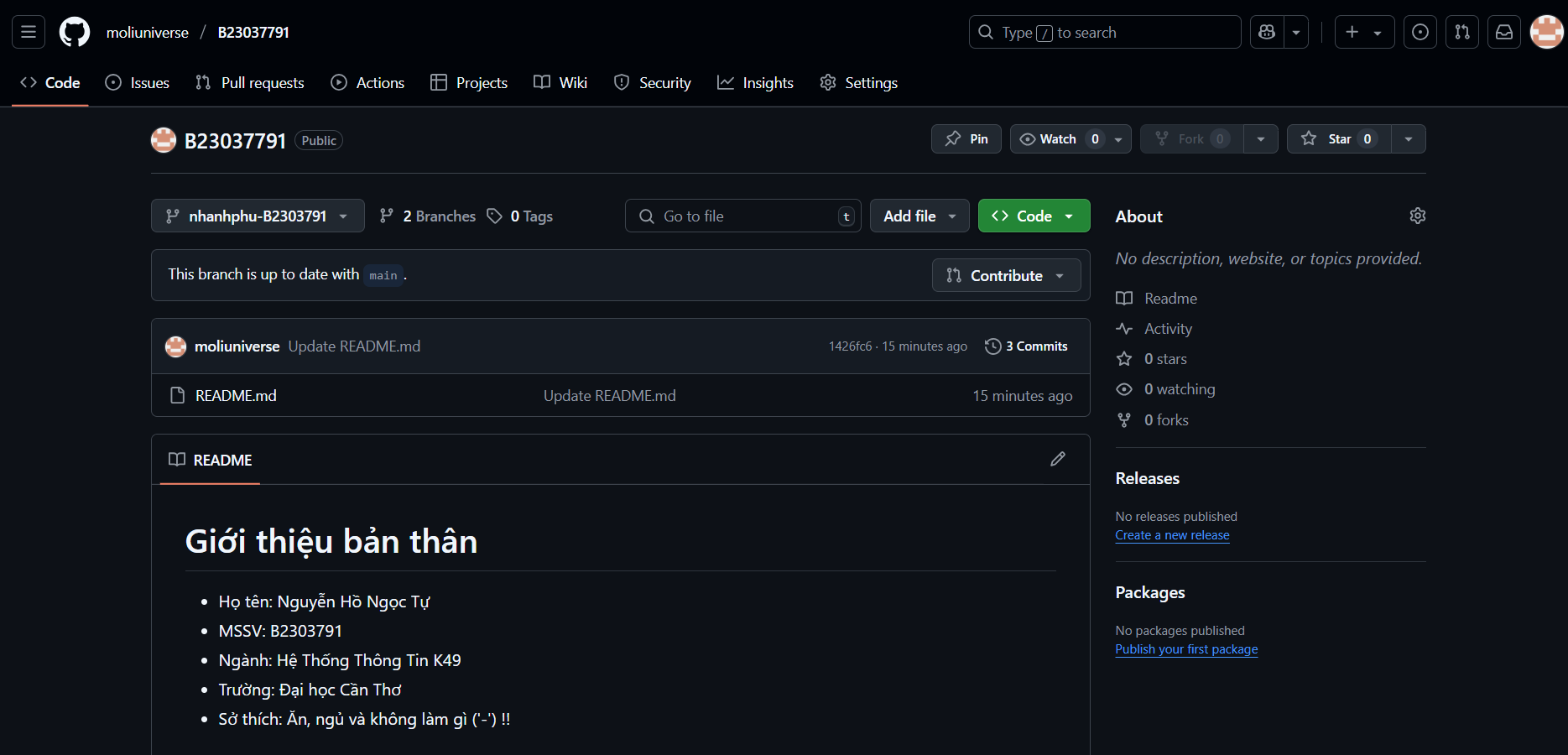
**Câu 4:**

**Branch** (nhánh) trong Git là một nhánh riêng biệt của mã nguồn — nơi bạn có thể làm việc, phát triển hoặc thử nghiệm mà không ảnh hưởng đến mã nguồn chính (thường là nhánh main hoặc master).

*Ví dụ: Khi bạn muốn thêm tính năng mới, bạn tạo một branch để làm việc độc lập. Sau khi hoàn thành và kiểm tra, bạn mới gộp lại vào nhánh chính (gọi là merge).*

Lợi ích của Branch:

* Làm việc song song trên nhiều tính năng.
* Tránh xung đột mã giữa các thành viên.
* Cho phép kiểm thử, sửa lỗi, phát triển riêng biệt.
* Giúp quản lý phiên bản linh hoạt và chuyên nghiệp.



*Link:* [*moliuniverse/B23037791 at nhanhphu-B2303791*](https://github.com/moliuniverse/B23037791/tree/nhanhphu-B2303791)

**Câu 5:**

1. **Commit**: git commit dùng để lưu lại thay đổi trong mã nguồn vào lịch sử của Git (trên máy tính cục bộ).

* Giống như bạn ghi lại một mốc lịch sử: “Tôi vừa sửa file này”, “Tôi vừa thêm tính năng mới”.
* Mỗi lần commit đều kèm theo một thông điệp mô tả (message) để biết bạn đã thay đổi gì.
  1. **Push**: git push là hành động đẩy (upload) các commit đã lưu trong máy lên GitHub (hoặc máy chủ Git từ xa).
* Nếu bạn chỉ commit mà chưa push, thì người khác không thấy các thay đổi của bạn.
* Push giúp chia sẻ công việc với đồng đội.
  1. **Pull:** git pull dùng để **lấy về và hợp nhất** những thay đổi mới nhất từ GitHub (hoặc máy chủ Git) vào máy của bạn.
* Dùng khi bạn làm việc nhóm để cập nhật những gì người khác đã làm.
* Tương đương với: git fetch + git merge.

Câu 6:

**Mở Terminal (hoặc Git Bash)**

Đi tới thư mục chứa **tập tin thực hành buổi 5**. Ví dụ:

bash

cd DuongDan/TenThuMuc/Buoi5

**✅ Bước 2: Khởi tạo Git (nếu chưa có)**

bash

git init

**✅ Bước 3: Kết nối đến GitHub repository của bạn**

Giả sử bạn đã tạo một repository trên GitHub có tên là thuchanh-buoi5, bạn nhập:

bash

git remote add origin https://github.com/ten-tai-khoan/thuchanh-buoi5.git

*(Nhớ thay ten-tai-khoan bằng tên GitHub của bạn nhé)*

**✅ Bước 4: Thêm tập tin vào vùng staging**

bash

git add .

*Lệnh này sẽ thêm tất cả tập tin trong thư mục vào danh sách để commit.*

**✅ Bước 5: Commit với nội dung mô tả**

bash

git commit -m "Upload bài thực hành buổi 5"

**✅ Bước 6: Push lên GitHub**

*Lần đầu tiên, bạn nên chỉ định tên nhánh chính là main:*

bash

git branch -M main

git push -u origin main

Câu 7:

**Lệnh git clone đầy đủ để tải repository về máy:**

bash

git clone https://github.com/moliuniverse/B23037791.git

**👉 Sau khi chạy lệnh trên:**

1. Một thư mục tên B23037791 sẽ được tạo trong máy.
2. Bạn có thể di chuyển vào thư mục đó:

bash

cd B23037791

1. Mọi nội dung trong repo (ví dụ: README.md) sẽ có sẵn để bạn chỉnh sửa, thêm file, commit hoặc push tiếp.

Câu 8:

**✅ 1. Chép tập tin vào thư mục vừa clone**

Giả sử bạn đã chạy lệnh sau và tải repo về máy:

bash

git clone https://github.com/moliuniverse/B23037791.git

Bạn sẽ có thư mục B23037791. Giờ hãy **chép tập tin thực hành buổi 4** vào thư mục đó:

📄 Tập tin:

CopyEdit

28-B2303791-NguyenHoNgocTu-Bai4

**Cách chép:**

* Thủ công: Kéo thả file vào thư mục B23037791.
* Hoặc dùng dòng lệnh nếu cần:

bash

CopyEdit

cp /duong\_dan/den\_file/28-B2303791-NguyenHoNgocTu-Bai4 B23037791/

**✅ 2. Thực hiện các lệnh Git**

Mở Terminal/Git Bash trong thư mục B23037791, chạy các lệnh sau:

bash

CopyEdit

git add .

git commit -m "upload du lieu"

git push

**✅ 3. Giải thích ý nghĩa từng lệnh**

|  |  |
| --- | --- |
| **Lệnh** | **Ý nghĩa** |
| git add . | Thêm tất cả các tập tin mới hoặc đã chỉnh sửa vào vùng staging để chuẩn bị commit. |
| git commit -m "upload du lieu" | Tạo một bản ghi (commit) với thông điệp "upload du lieu", ghi nhận lại thay đổi trong dự án. |
| git push | Gửi (đẩy) các commit vừa tạo từ máy bạn lên GitHub (repo trực tuyến). |

Câu 9:

**Chép tập tin thực hành buổi 3 vào thư mục repo**

Giả sử em có file tên là:

28-B2303791-NguyenHoNgocTu-Bai3.html

Và đã **clone** repo từ GitHub bằng lệnh:

bash

git clone https://github.com/moliuniverse/B23037791.git

Tiếp theo:

* **Chép file buổi 3** vào thư mục B23037791 vừa clone.

**✅ 2. Thực hiện các lệnh Git để đẩy lên GitHub**

**Mở terminal và gõ:**

bash

cd B23037791

git add .

git commit -m "Upload bài thực hành buổi 3"

git push

✅ Lúc này, file thực hành buổi 3 đã được đẩy lên GitHub.

**✅ 3. Thực hiện lệnh git pull**

bash

git pull

**👉 Ý nghĩa của git pull:**

* Lệnh git pull sẽ **kéo toàn bộ thay đổi mới nhất** từ repository trên GitHub **về máy tính cục bộ** và **hợp nhất** vào thư mục hiện tại.
* Dùng để đảm bảo máy của em luôn **cập nhật đầy đủ** nội dung mới từ GitHub (nhất là khi làm việc nhóm).

**✅ 4. Quan sát kết quả**

* Nếu **GitHub không có thay đổi nào mới** kể từ lần push, lệnh git pull sẽ báo:

vbnet

CopyEdit

Already up to date.

→ Nghĩa là máy em đã đồng bộ 100% với GitHub.

* Nếu **ai đó đã thêm file/commit trên GitHub**, lệnh git pull sẽ:
  + Tự động tải các thay đổi về.
  + Gộp nội dung mới vào thư mục của em.

**✅ 5. Nhận xét**

| **Trường hợp** | **Nhận xét** |
| --- | --- |
| ✅ File được upload thành công lên GitHub | Em sẽ thấy file 28-B2303791-NguyenHoNgocTu-Bai3.html xuất hiện trên trang repo |
| ✅ git pull không có thay đổi | Hệ thống đã đồng bộ, không có nội dung mới cần tải |
| ⚠️ Có xung đột (conflict) | Có thể xảy ra khi 2 người sửa cùng một file – lúc đó cần xử lý merge |

Câu 10:

Câu 11:

