Câu 1:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Câu 2:

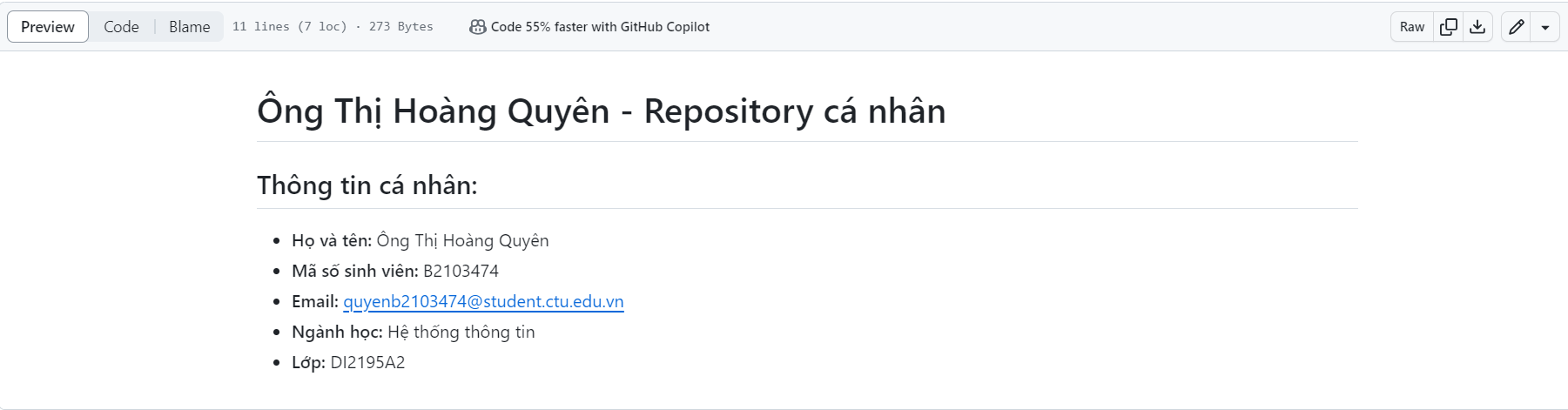
* Git và GitHub là hai công cụ quan trọng trong quản lý mã nguồn mở và dự án phần mềm.
* Lợi ích của Git:
* Cộng tác dễ dàng: Git giúp các nhà phát triển làm việc cùng nhau trên cùng một dự án một cách dễ dàng thông qua việc quản lý và tích hợp các nhánh (branches).
* Tốc độ và hiệu suất: Git được thiết kế để hoạt động nhanh chóng và hiệu quả, cho phép người dùng làm việc với các dự án lớn với tốc độ cao.
* Quản lý phiên bản: Git cho phép lưu trữ và quản lý các phiên bản của mã nguồn, giúp theo dõi sự thay đổi và khôi phục các phiên bản trước đó khi cần.
* Lịch sử sửa đổi: Git ghi lại thông tin chi tiết về các sửa đổi trong mã nguồn, bao gồm người thực hiện, thời gian và nội dung thay đổi, giúp theo dõi quá trình phát triển mã nguồn.
* Làm việc nhanh chóng: Git cho phép các nhánh (branches) song song phát triển, giúp nhóm làm việc độc lập mà không ảnh hưởng đến nhau.
* Hệ thống phân nhánh mạnh mẽ: Git cho phép các nhánh phát triển song song, giúp người dùng thực hiện các tính năng mới mà không ảnh hưởng đến mã nguồn chính.
* Khả năng làm việc ngoại tuyến: Git cho phép người dùng làm việc ngoại tuyến, lưu trữ thay đổi và đồng bộ thông tin khi kết nối mạng lại.
* Lợi ích của GitHup:
* Lưu trữ mã nguồn trực tuyến: GitHub cung cấp dịch vụ lưu trữ mã nguồn trực tuyến, giúp đội ngũ phát triển dễ dàng truy cập và quản lý mã nguồn từ bất kỳ đâu.
* Hợp tác dễ dàng: GitHub cung cấp công cụ quản lý dự án, theo dõi vấn đề, và các tính năng hợp tác giúp các nhà phát triển hiểu rõ hơn về quá trình phát triển dự án và tương tác với nhau.
* CI/CD và DevOps: GitHub tích hợp các dịch vụ liên quan đến Continuous Integration (CI), Continuous Delivery (CD) và quy trình DevOps, giúp tự động hóa quy trình phát triển phần mềm.
* Quản lý vấn đề thông minh: GitHub có hệ thống quản lý vấn đề tích hợp, cung cấp cơ chế theo dõi và quản lý vấn đề, giúp giải quyết vấn đề hiệu quả hơn trong quá trình phát triển.
* Tích hợp dịch vụ: GitHub tích hợp với các dịch vụ khác như Jira, Slack, và nhiều công cụ phần mềm khác để tăng cường hiệu suất và dễ dàng quản lý dự án.
* Quản lý vấn đề (Issue tracking): GitHub cho phép người dùng tạo và quản lý các vấn đề, bug và yêu cầu tính năng trong dự án một cách dễ dàng. Điều này giúp tăng cường tính tương tác và giải quyết vấn đề nhanh chóng.
* Hợp nhất mã (Pull requests): GitHub cho phép người dùng tạo các yêu cầu kéo (pull requests) để đề xuất và thảo luận về các thay đổi trong mã nguồn, giúp quản lý và hợp nhất các thay đổi một cách linh hoạt và minh bạch.
* Phân tích và xem lại mã nguồn: GitHub cung cấp các công cụ phân tích mã nguồn và xem lại, bao gồm việc hiển thị lịch sử thay đổi, thống kê hoạt động và xem lại các lần thay đổi cụ thể.
* Cộng đồng và mạng lưới: GitHub là một cộng đồng lớn của các nhà phát triển trên toàn thế giới, cho phép người dùng kết nối, học hỏi và chia sẻ kiến thức và kinh nghiệm với nhau.

Câu 3:

* Trong GitHub, "repositories" (kho chứa) là nơi lưu trữ tất cả các mã nguồn, tài liệu, hình ảnh, và dữ liệu khác liên quan đến dự án. Mỗi repository có thể được coi như một thư mục chứa toàn bộ dự án, và cũng bao gồm các lịch sử thay đổi, nhánh (branches), và các tính năng khác để quản lý dự án.
* GitHub cho phép mỗi người dùng tạo ra không giới hạn số lượng repositories công khai (public repositories). Tuy nhiên, cho các repositories riêng tư (private repositories), GitHub có các giới hạn về số lượng repositories và dung lượng lưu trữ:
* Dung lượng lưu trữ miễn phí tối đa cho các repositories riêng tư là 500 MB.
* Đối với các gói trả phí, dung lượng lưu trữ có thể được mở rộng lên tới hàng terabyte.
* GitHub hiện đang cho phép tải lên các tập tin vào repository với kích thước tối đa là 100MB:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động



https://raw.githubusercontent.com/hoangquyenabc/b2103474/main/README.md

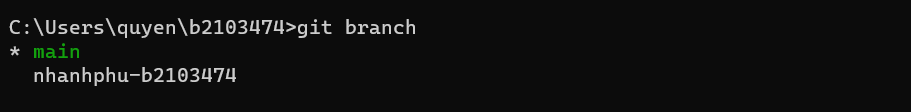
Câu 4:

* Trong Git, "branch" (nhánh) là một dạng tham chiếu đến một commit cụ thể. Branch giúp quản lý việc phát triển code song song mà không làm ảnh hưởng đến phiên bản chính của repository. Khi tạo một branch mới, bạn có thể phát triển tính năng mới, sửa đổi mã nguồn mà không ảnh hưởng đến phiên bản chính. Điều này giúp các nhà phát triển làm việc độc lập và có thể dễ dàng so sánh, kiểm tra và tích hợp các thay đổi mà không gây ảnh hưởng đáng kể đến mã nguồn chính.
* Tạo nhánh

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màu đen

Mô tả được tạo tự động

* Xem nhánh



Câu 5:

* Commit: là quá trình lưu trữ các thay đổi vào repository. Khi bạn thực hiện một commit, bạn đang tạo ra một bản ghi về trạng thái hiện tại của mã nguồn trong thư mục làm việc của bạn.
* Push: Là hành động gửi các commit từ repository cục bộ lên một repository trên server (như GitHub hay GitLab). Thông thường, sau khi bạn đã commit các thay đổi vào repository cục bộ, bạn sẽ push những thay đổi này lên repository chia sẻ để các thành viên khác trong nhóm có thể thấy và sử dụng.
* Pull: Là hành động lấy về (fetch) các thay đổi từ một repository server và áp dụng các thay đổi đó vào repository cục bộ của bạn. Quá trình pull thường được sử dụng để cập nhật mã nguồn cục bộ với phiên bản mới nhất trên server sau khi có sự thay đổi từ các thành viên khác trong nhóm hoặc từ các nhánh khác của repository.

Câu 8:

* git add . :
* Lệnh này thêm tất cả các thay đổi đã được sửa đổi hoặc mới tạo vào "index" (hoặc stage area) trong Git để chuẩn bị cho việc commit.
* Dấu chấm . ở đây có nghĩa là thêm tất cả các thay đổi từ thư mục hiện tại vào index.
* git commit -m "upload dữ liệu":
* Lệnh này tạo một commit từ các thay đổi đã được đặ sẵn trong index. Bằng cách sử dụng tùy chọn -m, bạn có thể cung cấp một thông điệp commit ngắn mô tả về nội dung của commit. Trong ví dụ này, message là "upload dữ liệu".
* Commit là hành động lưu trữ các thay đổi đã được thực hiện vào lịch sử của repository, làm cho chúng trở thành một phần của lịch sử phiên bản của dự án.
* Git push:
* Sau khi đã tạo commit, lệnh push gửi các commit từ repository cục bộ lên repository trên server (ví dụ như GitHub, GitLab, Bitbucket).
* Điều này đồng thời cũng cập nhật các thay đổi lên repository chia sẻ để các thành viên khác trong nhóm có thể thấy và sử dụng.