**GIT**

1. Tổng quan

Git là một hệ thống quản lý phiên bản phân tán **(Distributed Version Control System – DVCS)**

Hệ thống quản lý phiên bản (Version Control System - VCS) là một phần mềm giúp chúng ta lưu lại từng thay đổi của mã nguồn dự án, và giúp lấy lại được các phiên bản trước đó nếu cần.

Lợi ích của Git:

- Khôi phục lại trạng thái của dự án ở các thời điểm khác nhau trong quá khứ

- Biết được ai đã thực hiện các thay đổi trên dự án, và đã thay đổi những gì

- Dễ dàng khôi phục lại các nội dung mã nguồn bị xóa

- Dễ dàng so sánh những thay đổi của dự án theo các mốc thời gian

Hệ thống quản lý phiên bản gồm 3 loại gồm:

- Hệ thống quản lý phiên bản cục bộ

-> Lưu trữ mỗi phiên bản trong 1 thư mục. Không hỗ trợ trên mỗi trường cộng tác nhiều người.

- Hệ thống quản lý phiên bản tập trung

-> Máy chủ chứa các phiên bản của thư mục dự án và danh sách các máy khách được phép thay đổi thư mục dự án trên máy chủ. Máy khách sẽ lấy từ máy chủ về cập nhật lại các thay đổi về máy chủ.

-> Hạn chế của hệ thống quản lý phiên bản tập trung là nếu máy chủ bị trục trặc sẽ dẫn tới gián đoạn việc cập nhật phiên bản của các thành viên, hoặc bị mất toàn bộ các phiên bản của thư mục dự án trên server.

- Hệ thống quản lý phiên bản phân tán.\

-> Hệ thống quản lý phiên bản phân tán cũng cho phép một máy client có thể liên kết với nhiều kho chứa ở xa cùng lúc, do vậy bạn có thể cùng lúc cộng tác với nhiều nhóm làm việc khác nhau.

-> Cho phép client chép toàn bộ kho chứa gồm cả lịch sử các phiên bản. Kh server trục trặc có thể sử dụng kho chứa cửa client để khôi phục lại cho server

2) Setup môi trường lập trình

a) Định danh người dùng

Lập trình viên phải định danh tên để biết ai code

Độ ưu tiên phạm vi:

local > global > system.

Local:

  + Phạm vi này áp dụng cho một kho lưu trữ cụ thể.

Global:

 + Phạm vi này áp dụng cho tài khoản người dùng hiện tại và cho tất cả kho lưu trữ mà người dùng đó làm việc.

System:

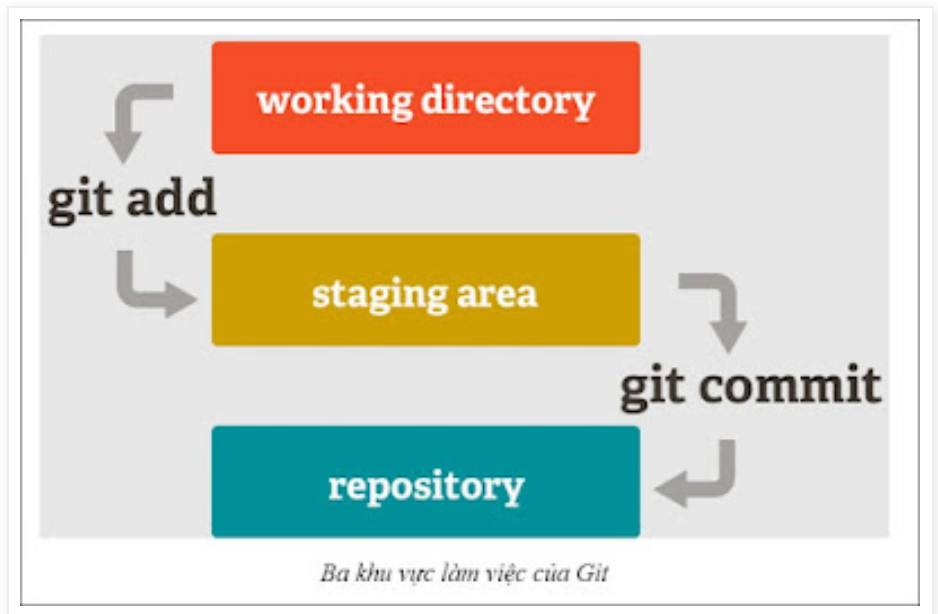
+ Phạm vi này áp dụng cho tất cả người dùng và tất cả kho lưu trữ trên hệ thống.

b) Cấu hình định danh người dùng

git config --global --unset user.name

git config --global --unset user.email

c) Khu vực làm việc



1 **Working Directory (Thư mục làm việc)**

* Đây là nơi bạn thực hiện chỉnh sửa mã nguồn.
* Các tệp trong thư mục này có thể ở ba trạng thái: **untracked (chưa được theo dõi)**, **modified (đã chỉnh sửa)** hoặc **staged (đã được đánh dấu để commit)**.
* Khi bạn tạo, chỉnh sửa hoặc xóa file trong thư mục làm việc, Git sẽ ghi nhận sự thay đổi.

2 **Staging Area (Khu vực tạm thời / chỉ mục – Index)**

* Đây là nơi Git theo dõi các thay đổi đã được đánh dấu để commit.
* Bạn sử dụng lệnh git add để đưa file từ **Working Directory** vào **Staging Area**.
* Điều này giúp bạn có thể chuẩn bị một commit với các thay đổi cụ thể.
* Bạn có thể kiểm tra trạng thái bằng lệnh:

3 **Git Repository (Kho lưu trữ – Local Repository)**

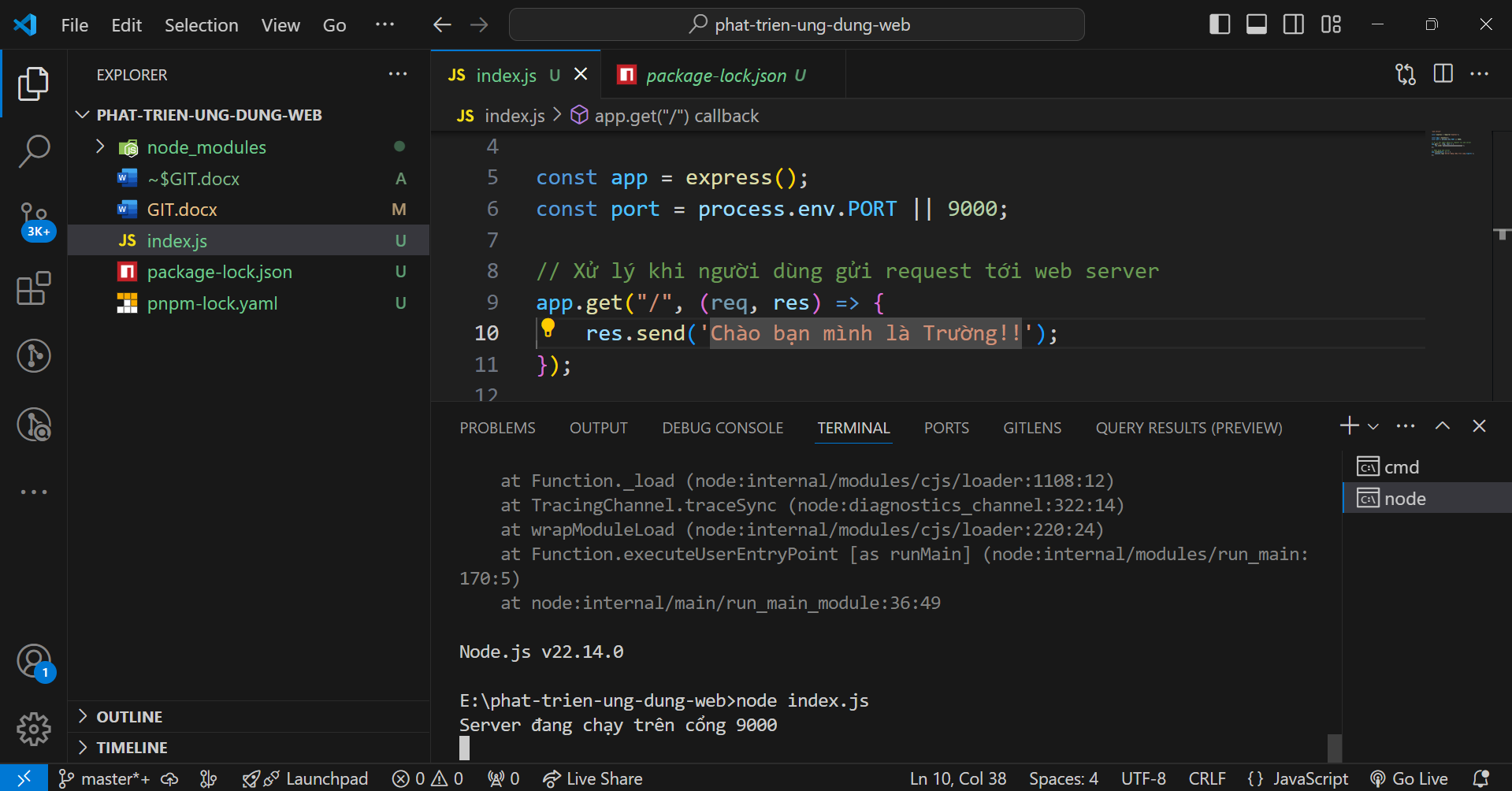
* Đây là nơi Git lưu trữ tất cả lịch sử commit của dự án.
* Khi bạn chạy lệnh git commit, Git sẽ lưu trạng thái của các tệp trong **Staging Area** vào kho lưu trữ.
* Git sử dụng một thư mục ẩn .git/ để quản lý toàn bộ lịch sử thay đổi.

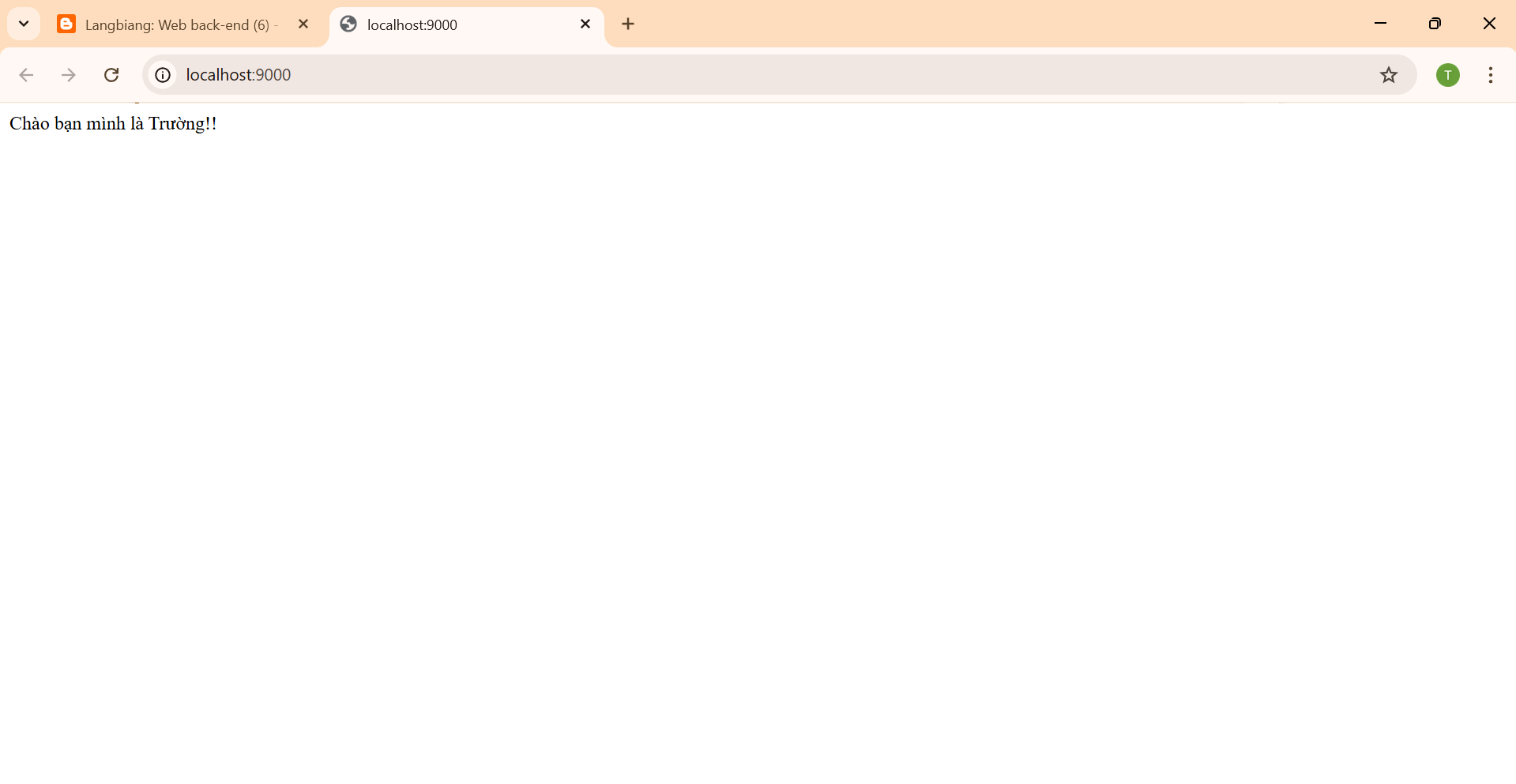
Lệnh:

git init -> working space -> git add -> staging area -> git commit -> repository

II) Backend

1. Nodejs, Npm
2. Global, Dependencies,devDependencies
3. Javascript
   * Bất đồng bộ
   * CallBack
   * Arrow Function





1.6 Bài tập

Câu 1.1 Các đặc điểm của hệ thống quản lý phiên bản cục bộ. Phát biểu nào không đúng?

A. Có thể thực hiện thủ công

B. Không hỗ trợ trong môi trường cộng tác nhiều người

C. Có thể dùng phần mềm để quản lý phiên bản kiểu cục bộ

D. Các phiên bản của dự án được lưu tập trung trên một máy server

Câu 1.2 Các đặc điểm của hệ thống quản lý phiên bản tập trung. Phát biểu nào không đúng?

A. Các phiên bản của dự án được lưu tập trung trên máy server

B. Các máy client sẽ chứa tất cả các phiên bản của thư mục dự án cùng với lịch sử thay đổi

C. Máy client không thể tải phiên bản của dự án về, khi máy server không hoạt động

D. Hỗ trợ làm việc cộng tác nhiều người

Câu 1.3 Các đặc điểm của hệ thống quản lý phiên bản phân tán. Phát biểu nào không đúng?

A. Các máy client sẽ chứa toàn bộ các phiên bản của dự án, cùng lịch sử thay đổi

B. Hỗ trợ làm việc cộng tác nhiều người

C. Các phiên bản của dự án được lưu trên máy server

D. Bạn không thể tạo và lưu phiên bản khi không có kết nối mạng tới máy server

Câu 1.4 Tìm trang chủ của các phần mềm quản lý phiên bản.

|  |  |
| --- | --- |
| Tên phần mềm | Trang chủ |
| RCS | https://www.gnu.org/software/rcs/ |
| CVS | https://www.nongnu.org/cvs/ |
| Subversion (SVN) | https://subversion.apache.org/ |
| Git | https://git-scm.com/ |
| Mercurial | https://www.mercurial-scm.org/ |
| Darcs | https://darcs.net/ |

Bài tập 2.1 Thực hành các cài đặt trong bài học.

2.2 Lệnh nào được sử dụng để nhúng Git vào thư mục dự án?

A. git init

B. git --init

C. git initialize

D. git embed

2.3 Lệnh nào sử dụng để kiểm tra trên máy tính đã có phần mềm Git hay chưa?

A. git ver

B. git version

C. git -v

D. git --ver

2.4 Trong Git, kho lưu trữ (repo, repository) là gì?

A. Là thư mục dự án

B. Là thư mục dự án đã được nhúng Git

C. Là thư mục cài đặt phần mềm Git

D. Là thư mục *.git* (trong thư mục dự án)

Bài tập 3.1 Bài tập tình huống “Quản lý cấu hình Git cho nhiều dự án”

Giả sử bạn là Nguyễn Văn Tèo, một lập trình viên đang làm việc tại công ty TeoTech. Ngoài công việc chính, bạn còn tham gia phát triển một dự án mang tên VienVong (viển vông) cùng với một nhóm bạn vào thời gian rảnh.

Để quản lý mã nguồn hiệu quả, bạn sử dụng Git cho cả hai dự án. Tuy nhiên, bạn muốn đảm bảo rằng:

- Khi làm việc tại công ty TeoTech, các commit của bạn sẽ được gắn với thông tin định danh của công ty (tên và email công ty).

- Khi làm việc với dự án "viển vông", các commit của bạn sẽ được gắn với thông tin định danh cá nhân.

***Yêu cầu:***

- Tạo hai thư mục dự án riêng biệt: TeoTechProject (cho dự án công ty). VienVongProject (cho dự án cá nhân)

- Khởi tạo kho lưu trữ Git trong mỗi thư mục dự án.

- Cấu hình Git sao cho:

  + Khi commit trong dự án TeoTechProject, Git sử dụng thông tin định danh: Tên: "Nguyen Van Teo"; Email: nvteo@teotech.com

  + Trong commit trong dự án VienVongProject, Git sử dụng thông tin định danh: Tên: "Nguyen Van Teo"; Email: nvteo@gmail.com

***Mục tiêu:***

Bài tập này giúp các bạn hiểu rõ hơn về cách cấu hình Git cho nhiều dự án với các thông tin định danh khác nhau, đồng thời rèn luyện kỹ năng làm việc với Git trong môi trường thực tế.

Câu 3.2 Phạm vi cấu hình định danh trong Git là gì? Phát biểu nào sau đây không đúng?

A. Phạm vi *system*áp dụng cho tất cả người dùng và kho lưu trữ trên hệ thống.

B. Phạm vi *global* áp dụng cho tài khoản người dùng hiện tại và tất cả kho lưu trữ mà người dùng đó làm việc.

C. Phạm vi local áp dụng cho một kho lưu trữ cụ thể và có độ ưu tiên thấp nhất.

D. Phạm vi local áp dụng cho một kho lưu trữ cụ thể và có độ ưu tiên cao nhất.

Bài tập 4.1 Thực hành lại các lệnh, tạo các thư mục, tập tin như trong bài học. Xong

Câu 4.2 Ba khu vực làm việc chính của Git là gì? Phát biểu nào sau đây không đúng?

A. Thư mục làm việc (working directory) là nơi bạn chỉnh sửa tập tin trực tiếp.

B. Khu tạm (staging area) là nơi lưu trữ lịch sử phiên bản của dự án.

C. Kho chứa (repository) là nơi lưu trữ dữ liệu chính thức của dự án.

D. Lệnh git add được sử dụng để chuyển các thay đổi từ Thư mục làm việc sang Khu tạm.

Câu 4.3 Thư mục làm việc (working directory) trong Git là gì? Phát biểu nào sau đây không đúng?

A. Là thư mục thực tế trên máy tính, nơi bạn làm việc trực tiếp với các tập tin của dự án.

B. Là nơi bạn tạo, sửa, hoặc xóa tập tin trước khi đưa thay đổi vào .git.

C. Là nơi lưu trữ lịch sử phiên bản và dữ liệu chính thức của dự án.

D. Phản ánh các thay đổi so với commit cuối cùng, giúp bạn quyết định những gì cần thêm vào Khu tạm (staging area).

Bài tập 5.1 Thực hành các nội dung trong bài học

Câu hỏi 5.2 Khu tạm (staging area) trong Git là gì? Phát biểu nào sau đây **không** đúng?

A. Khu tạm là một khu vực trung gian giữa Thư mục làm việc (working directory) và Kho chứa (repository).

B. Khu tạm là một thư mục vật lý trong Thư mục làm việc.

C. Khu tạm cho phép bạn chọn lọc và tổ chức các thay đổi bạn muốn đưa vào commit tiếp theo.

D. Lệnh git rm --cached <tên\_tập\_tin> được sử dụng để gỡ bỏ tên tập tin ra khỏi Khu tạm.

Câu hỏi 5.3 gitignore là gì? Phát biểu nào sau đây **không** đúng?

A. gitignore giúp thêm các tập tin không liên quan (ví dụ: node\_modules, tập tin log) vào kho chứa.

B. gitignore là một cơ chế trong Git, cho phép bạn chỉ định các tập tin hoặc thư mục mà Git sẽ bỏ qua khi thực hiện các lệnh như git add hoặc git status.

C. gitignore được thực hiện thông qua tập tin có tên là .gitignore, thường đặt ở thư mục gốc của dự án.

D. Mục đích của gitignore là loại bỏ các tập tin không cần thiết khỏi lịch sử phiên bản, như tập tin tạm, tập tin nhị phân, hoặc thư viện phụ thuộc.

Câu hỏi 5.4 Kho chứa (repository) trong Git là gì? Phát biểu nào sau đây **không** đúng?

A. Kho chứa là nơi lưu trữ tất cả các tập tin và lịch sử thay đổi của một dự án.

B. Thư mục .git trong dự án chính là Kho chứa.

C. Lệnh git commit được sử dụng để chuyển tập tin từ Thư mục làm việc (working directory) sang Kho chứa.

D. Kho chứa cục bộ (local repository) là kho chứa được lưu trữ trên một máy chủ ở xa.

Câu 5.5 Commit trong Git là gì? Phát biểu nào sau đây **không** đúng?

A. Commit là lệnh trong Git dùng để lưu các thay đổi từ Khu tạm vào Kho chứa.

B. Commit tạo ra một "ảnh chụp" (snapshot) mới trong lịch sử phiên bản.

C. Commit dùng để chuyển tập tin từ Kho chứa sang Khu tạm.

D. Mỗi commit đi kèm một thông điệp (message) mô tả thay đổi.

Bài tập 5.1 Tải và cài đặt Nodejs, NPM trên máy tính. Kiểm tra để đảm bảo Nodejs và NPM sẵn sàng sử dụng.

Câu 5.2 Nodejs là gì? Phát biểu nào không đúng?

A. Nodejs là môi trường phát triển ứng dụng

B. Nodejs là một ngôn ngữ lập trình

C. Nodejs là nền tảng phát triển ứng dụng

D. Có thể sử dụng Nodejs để tạo web server

Câu 5.3 NPM là gì? Phát biểu nào không đúng?

A. NPM là viết tắt của Node Package Manager

B. NPM là công cụ để quản lý các thư viện dùng trong một ứng dụng chạy trên nền Nodejs

C. NPM được cài đặt mặc định khi cài đặt Nodejs

D. NPM gồm 3 thành phần: website, GUI và registry

Bài tập 6.1 Thực hiện các cài đặt, cấu hình trong bài học.

Bài tập 6.2 Làm sao biết được một gói cài đặt ở chế độ cục bộ (locally) hay toàn cục (globally)? Khi cài đặt các gói (package) ở chế độ cục bộ và toàn cục (globally) thì các gói sẽ được lưu ở đâu? Làm sao bạn biết được?

Bài tập 6.3 Sự khác nhau khi cài đặt gói ở chế độ dependencies và devDependencies?

Câu 6.4 Trong ứng dụng chạy trên nền Nodejs, tập tin package.json dùng để làm gì? Phát biểu nào không đúng?

A. Chứa thông tin mô tả về ứng dụng

B. Quản lý các phụ thuộc

C. Chỉ ra tập tin JavaScript đầu tiên được thực thi

D. Chứa mã để tạo giao diện ứng dụng

Câu 6.5 PNPM là gì? Phát biểu nào không đúng?

A. Giúp tiết kiệm dung lượng đĩa

B. Có thể thay thế hoặc bổ sung cho NPM

C. Là viết tắt của Performant Node Package Manager

D. Không thể thay thế hoặc bổ sung cho NPM

Câu 6.6 Express trong Node.js được sử dụng để làm gì?

A. Tạo ứng dụng web và API

B. Quản lý cơ sở dữ liệu

C. Thiết kế giao diện người dùng

D. Kiểm tra hiệu suất ứng dụng

Câu 6.7 Nodemon trong Node.js được sử dụng để làm gì?

A. Kết nối với cơ sở dữ liệu

B. Tạo giao diện người dùng cho ứng dụng

C. Quản lý các gói phụ thuộc

D. Tự động khởi động lại server khi mã thay đổi

Bài tập 7.1 Thực hiện các cài đặt trong bài học.

Câu 7.2 Trong môi trường phát triển ứng dụng Nodejs, phát biểu nào không đúng khi nói về gói cục bộ?

A. Gói cục bộ được cài đặt trong thư mục node\_modules của dự án

B. Gói cục bộ chỉ có thể được sử dụng trong dự án mà nó được cài đặt

C. Gói cục bộ được quản lý thông qua file package.json và có thể sử dụng các phiên bản khác nhau giữa các dự án

D. Gói cục bộ luôn được cài đặt toàn cục trên hệ thống để tất cả các dự án đều có thể truy cập

Câu 7.3 Trong môi trường phát triển ứng dụng Nodejs, phát biểu nào không đúng khi nói về gói toàn cục?

A. Gói toàn cục luôn được liệt kê trong tập tin package.json của dự án

B. Gói toàn cục thường được sử dụng cho các công cụ dòng lệnh (CLI) như nodemon

C. Gói toàn cục được lưu trong thư mục toàn cục của hệ thống và có thể truy cập từ bất kỳ dự án nào

D. Gói toàn cục được cài đặt bằng lệnh *npm install -g <package-name>*

Câu 7.4 Trong môi trường phát triển ứng dụng Nodejs, phát biểu nào không đúng khi nói về kiểu cài đặt dependencies?

A. Các gói trong dependencies được cài đặt cục bộ trong dự án bằng lệnh *npm install <package-name>*

B. Các gói trong dependencies là các phụ thuộc cần thiết để ứng dụng chạy trong môi trường triển khai, sản xuất (production)

C. Các gói trong dependencies chỉ được sử dụng trong giai đoạn phát triển và không cần thiết khi triển khai ứng dụng (production)

D. Các gói trong dependencies được liệt kê trong tập tin package.json và tự động cài đặt khi chạy npm install

Câu 7.5 Trong môi trường phát triển ứng dụng Nodejs, phát biểu nào không đúng khi nói về kiểu cài đặt devDependencies?

A. Các gói trong devDependencies được cài đặt bằng lệnh *npm install <package-name> --save-dev*

B. Các gói trong devDependencies là các phụ thuộc cốt lỗi để ứng dụng chạy trong môi trường triển khai (production)

C. Các gói trong devDependencies chỉ cần thiết trong giai đoạn phát triển hoặc kiểm thử, không cần cho môi trường triển khai (production)

D. Các gói trong devDependencies không được cài đặt khi chạy *npm install –production*

Bài tập 9.1 Cài đặt các ví dụ trong bài học.

Câu 9.2: Hàm mũi tên (arrow function) là gì? Phát biểu nào sau đây không đúng về hàm mũi tên?

A. Hàm mũi tên có cú pháp ngắn gọn, giúp viết mã dễ đọc hơn.

B. Hàm mũi tên có this (ngữ cảnh) riêng, không kế thừa từ phạm vi bên ngoài.

C. Hàm mũi tên không thể được dùng làm hàm tạo (constructor).

D. Hàm mũi tên không có biến arguments để truy cập danh sách tham số.

Câu 9.3: Lập trình bất đồng bộ là gì? Phát biểu nào sau đây không đúng về lập trình bất đồng bộ?

A. Lập trình bất đồng bộ cho phép chương trình thực hiện nhiều tác vụ cùng lúc mà không cần chờ đợi tác vụ trước đó hoàn thành.

B. Lập trình bất đồng bộ đặc biệt hữu ích khi xử lý các tác vụ tốn thời gian như yêu cầu mạng, đọc/ghi tập tin hoặc tương tác với cơ sở dữ liệu.

C. Callback là một kỹ thuật bất đồng bộ, nó sẽ làm cho code dễ đọc và dễ bảo trì hơn khi code có nhiều callback lồng nhau.

D. Async/await cho phép viết mã bất đồng bộ trông giống như mã đồng bộ, giúp mã trở nên dễ đọc và dễ hiểu hơn.