**Web back-end (1) - Một số khái niệm**

Câu 1.1 Liệt kê 3 gói thuê bao (tên gói, giá tiền) của 3 nhà cung cấp kết nối Internet tại Việt Nam.

- Viettel: 165.000đ/tháng

- VNPT: 189.000đ/tháng

- FPT Telecom: 190.000đ/tháng

Câu 1.2:

Khả năng rằng các lập trình viên đã sử dụng Node.js ở phía máy chủ

Câu 1.3 Phát biểu nào không đúng khi nói về web, trang web và website?

A. Web là hệ thống thông tin toàn cầu

B. Web được tạo thành nhờ các trang web và sự liên kết giữa chúng

C. Một trang web chứa nhiều website

D. Một website chứa nhiều trang web có liên quan đến nhau

🡪 C

Câu 1.4 Phát biểu nào không đúng khi nói về website và ứng dụng web?

A. Không có sự phân biệt rõ ràng giữa website và ứng dụng web

B. Một website và ứng dụng web có một địa chỉ URL để truy cập

C. Website và ứng dụng web hoạt động dựa trên mô hình client-server

D. Chỉ cần sử dụng HTML và CSS là có thể tạo ra được ứng dụng web

🡪 D

**Web back-end (2) - Tổng quan về phát triển ứng dụng web**

2.1. Tìm trang (web) chủ của các CMS trong lĩnh vực phát triển ứng dụng web:

1. Thương mại điện tử/bán hàng:

WordPress: https://wordpress.org​

Wix: https://www.wix.com​

Joomla: https://www.joomla.org

Drupal: https://www.drupal.org​

Magento: https://business.adobe.com/products/magento/magento-commerce.html​

Shopify: https://www.shopify.com​

2. Hệ thống quản lý học tập:

Moodle: https://moodle.org​

3. Blogs, thông tin, giới thiệu:

Blogger: https://www.blogger.com​

WordPress: <https://wordpress.org>

2.2. Tìm các trang (web) chủ của các framework và thư viện hỗ trợ trong việc phát triển ứng dụng web:

1. Frontend:

jQuery: https://jquery.com​

Bootstrap: https://getbootstrap.com​

D3.js: https://d3js.org​

React: <https://react.dev>​

Angular: https://angular.io​

Vue.js: https://vuejs.org​

Svelte: https://svelte.dev​

2. Backend:

Express.js (Node.js): https://expressjs.com​

Django (Python): https://www.djangoproject.com​

Flask (Python): https://flask.palletsprojects.com​

ASP.NET (C#): https://dotnet.microsoft.com/apps/aspnet

Spring Boot (Java): https://spring.io/projects/spring-boot​

Laravel (PHP): https://laravel.com​

Ruby on Rails (Ruby): https://rubyonrails.org

Gin (Go): https://gin-gonic.com​

Echo (Go): <https://echo.labstack.com>

2.3 Có 3 cách để tạo ra một ứng dụng web? Phát biểu nào không đúng?

A. Sử dụng CMS

B. Sử dụng framework và thư viện hỗ trợ

C. Viết thuần bằng ngôn ngữ lập trình mà không sử dụng framework, thư viện hỗ trợ

D. Sử dụng trình duyệt web

🡪 D

2.4 Những ưu điểm khi sử dụng CMS để tạo ra ứng dụng web. Phát biểu nào không đúng?

A. Cần kiến thức lập trình chuyên sâu

B. Tối ưu SEO

C. Dễ sử dụng, dễ quản lý, tiết kiệm thời gian

D. Chi phí phát triển ứng dụng thấp

🡪 A

2.5 Những ưu điểm khi sử dụng framework và thư viện để tạo ra ứng dụng web. Phát biểu nào không đúng?

A. Tăng tốc độ phát triển

B. Cải thiện chất lượng viết mã

C. Giảm khả năng bảo mật

D. Dễ dàng bảo trì và mở rộng

🡪 C

**Web back-end (3) - Web server**

Câu 3.2 Một số phần mềm web server phổ biến. Phát biểu nào không đúng?

A. Apache

B. Nginx

C. SSI

D. Nodejs

🡪C 🡪 SSI để nhúng

Câu 3.3 Các chức năng của Web server là gì. Phát biểu nào không đúng?

A. Lưu trữ các tập tin của ứng dụng web

B. Nhận request từ client

C. Xử lý request

D. Nhận response từ máy client

🡪 D

Bài tập 3.4 Thực hiện một số cấu hình cho Nginx:

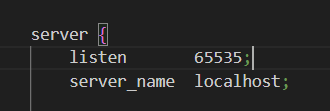
- (a) Lắng nghe trên cổng 65535

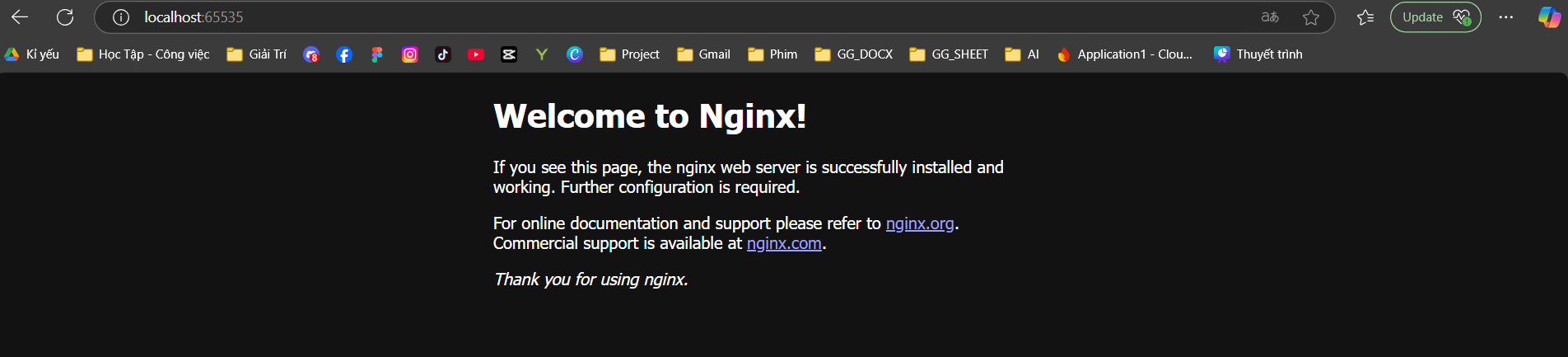
- (b) Trỏ web root về một thư mục khác ổ đĩa C:\ (ví dụ: D:\TeoShop)

- (c) Khi client gửi request tới web root, tập tin mặc định trả về là index.py, tạo thêm tập tin index.py để kiểm tra.

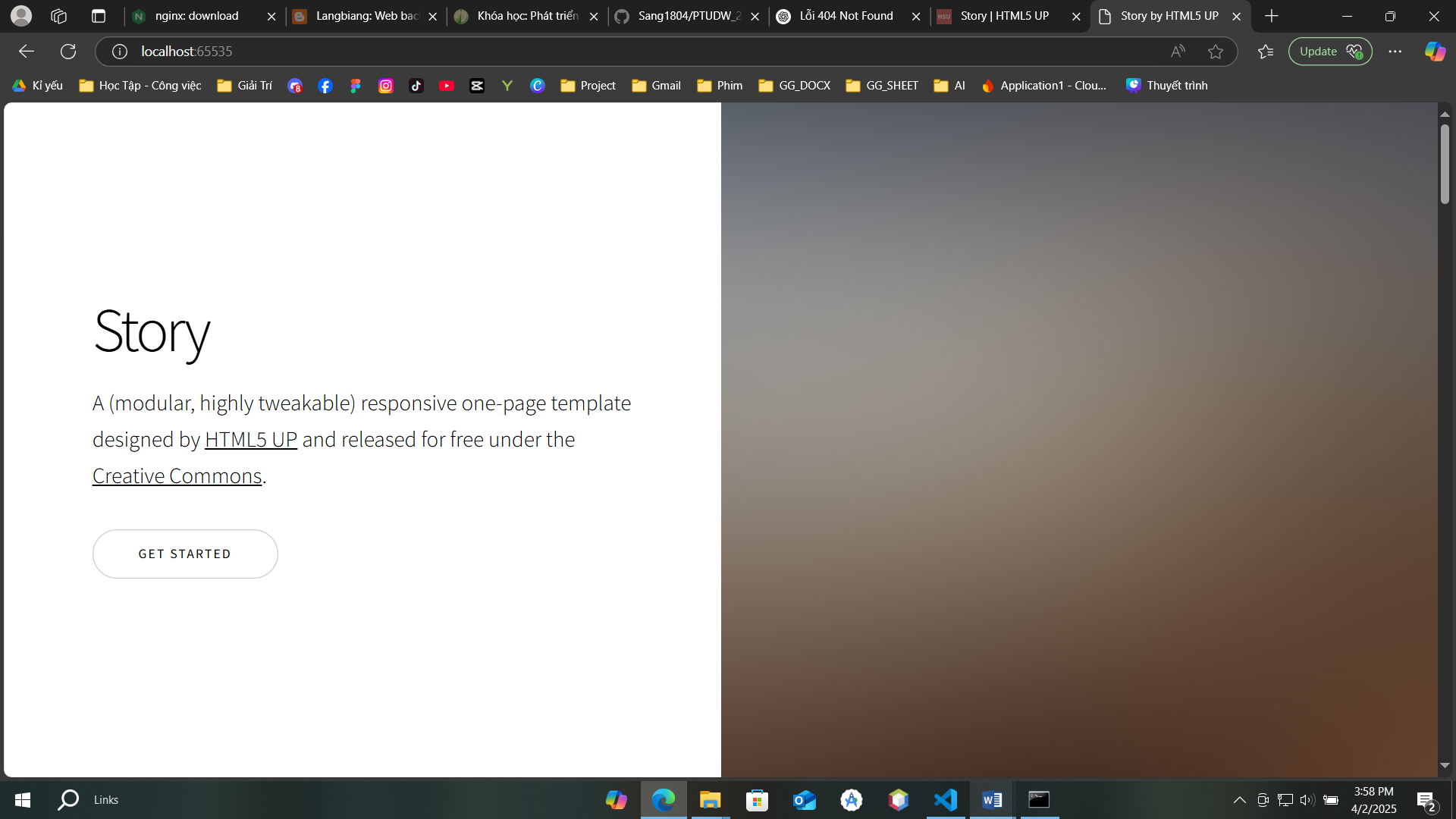
- (d) Lắng nghe trên cổng 65536 được không? Tại sao

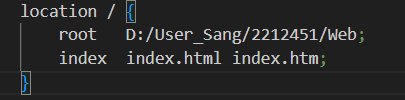
a)



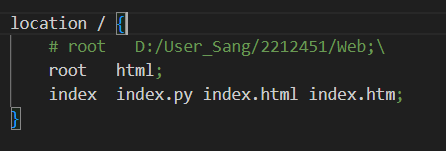


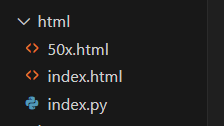
b)





c)





nó sẽ mở ra file index.py nhưng vì nginx không tự động hiểu .py nên nó sẽ trả về dạng file tải xuống

d) nó sẽ báo lỗi vì vượt ngoài phạm vi hợp lệ:

216 – 1 = 65535

**Web back-end (4) - Ứng dụng web bán hàng**

4.2 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

🡪 A

4.3 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 --version hoặc git -v

D. git –ver

🡪 C

4.4 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)

🡪D

4.5 Bạn có thể sử dụng các công cụ sau để viết mã. Phát biểu nào không đúng?

A. Text editor

B. MS Word

C. IDE, Code Editor

D. Chatbot (Gemini), Trang web (<https://codepen.io/>)

🡪 B

**Git thực hành (1) - Hệ thống quản lý phiên bản**

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

🡪 D

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

🡪B

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

🡪 D

**Git thực hành (2) - Tổng quan về Git**

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

🡪 A

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

🡪 C

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

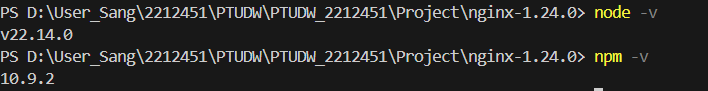
C. Là thư mục cài đặt phần mềm Git

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

🡪 D

**Web back-end (5) - Nodejs và NPM**

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

🡪 B

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

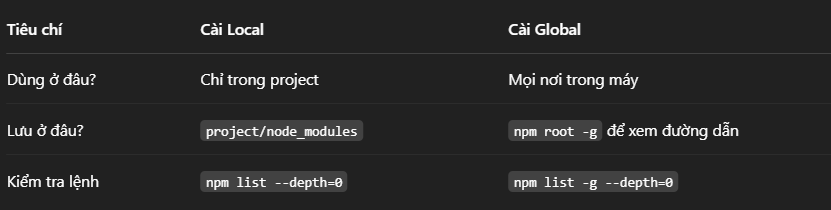
C. NPM được cài đặt mặc định khi cài đặt Nodejs

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

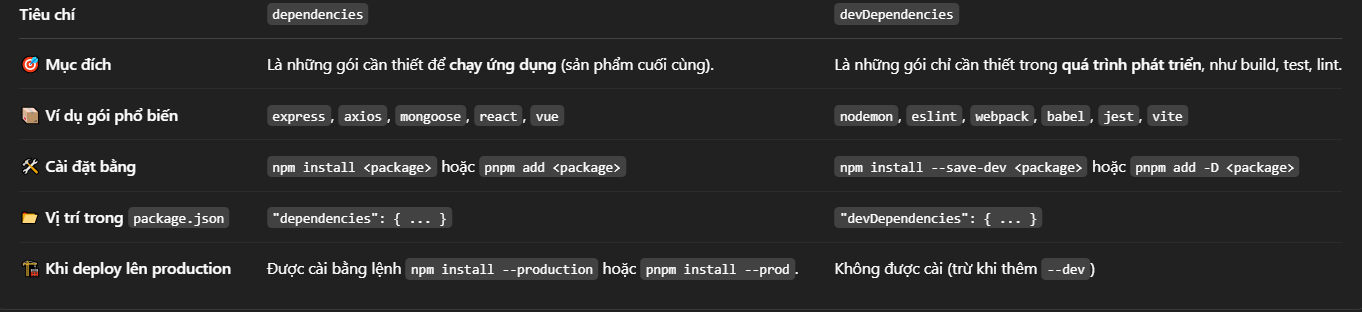
🡪 D 🡪 Không có GUI mà là CLI

**Web back-end (6) - Cấu hình Web server trong Nodejs**

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

🡪 D

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

🡪 D

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

🡪 A

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

🡪 D

**Web back-end (7) - Local và Global, Dependencies và devDependencies**

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

🡪 D

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>

🡪 A

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

🡪 C

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

🡪 B

**Web back-end (8) - Một số chủ đề JavaScript (1)**

Câu 8.2: Lập trình đồng bộ trong JavaScript là gì? Phát biểu nào sau đây không đúng?

A. Lập trình đồng bộ không chặn luồng, cho phép các lệnh tiếp theo chạy ngay cả khi tác vụ chưa hoàn tất.

B. Các lệnh được thực thi tuần tự, theo thứ tự từ trên xuống dưới.

C. Mỗi lệnh phải hoàn thành trước khi lệnh tiếp theo được thực thi.

D. Phù hợp với các tác vụ đơn giản như tính toán cơ bản không cần chờ đợi.

🡪 A

Câu 8.3: Hàm ẩn danh (anonymous function) trong JavaScript là gì? Phát biểu nào sau đây không đúng?

A. Hàm ẩn danh có thể truy cập biến trong phạm vi bao quanh nhờ closure.

B. Hàm ẩn danh là hàm không có tên khi được định nghĩa.

C. Hàm ẩn danh thường được gán vào biến hoặc truyền làm tham số cho hàm khác.

D. Hàm ẩn danh được "nâng lên" (hoisted) giống như hàm khai báo (function declaration).

🡪 D

**Web back-end (9) - Một số chủ đề JavaScript (2)**

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.

🡪 B

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.

🡪 C

**Web back-end (10) - Hàm callback**

Câu 10.2 Hàm callback trong JavaScript là gì? Phát biểu nào sau đây không đúng?

A. Hàm callback là một hàm được truyền vào một hàm khác như một tham số.

B. Hàm callback thường được sử dụng để xử lý các tác vụ bất đồng bộ.

C. Hàm callback có thể là hàm ẩn danh, hàm mũi tên hoặc hàm đã được định nghĩa trước.

D. Hàm callback được thực thi ngay lập tức sau khi hàm "cha" bắt đầu thực hiện.

🡪 D

Câu 10.3 Callback hell trong JavaScript là gì? Phát biểu nào sau đây không đúng?

A. Callback hell xảy ra khi có quá nhiều hàm callback lồng nhau, khiến mã nguồn khó đọc và bảo trì.

B. Callback hell thường xuất hiện khi xử lý các tác vụ bất đồng bộ phức tạp.

C. Callback hell là một cách hiệu quả để quản lý các tác vụ bất đồng bộ trong JavaScript.

D. Callback hell có thể được giải quyết bằng Promise hoặc async/await.

🡪 C

**Git thực hành (3) - Cấu hình định danh người dùng**

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

Dự án công ty (TeoTechProject):

- git config user.name "Nguyen Van Teo"

- git config user.email [nvteo@teotech.com](mailto:nvteo@teotech.com)

🡪 nếu sử dụng theo yêu cầu thì thêm – global 🡪 Cấu hình chung cho mọi repo

Dự án cá nhân (VienVongProject):

- git config user.name "Nguyen Van Teo"

- git config user.email [nvteo@gmail.com](mailto:nvteo@gmail.com)

🡪 Kiểm tra lại bằng git config --list

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.

🡪 C

**Git thực hành (4) - Các khu vực làm việc của Git**

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.

🡪 B

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).

🡪 C

**Git thực hành (5) - Các khu vực làm việc của Git**

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.

🡪 B

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.

🡪 A

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.

🡪 D

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.

🡪 C

**Web back-end (11) - Lập trình giao diện**

Câu 11.2 Free HTML template là gì? Phát biểu nào sau đây không đúng?

A. Free HTML template là các mẫu giao diện web được thiết kế sẵn, bao gồm HTML, CSS, và đôi khi JavaScript, được cung cấp miễn phí.

B. Free HTML template thường có giao diện đẹp, cấu trúc cơ bản (header, footer, sidebar), và phù hợp cho nhiều loại dự án như blog, portfolio, hoặc trang doanh nghiệp nhỏ.

C. Sử dụng Free HTML template tốn nhiều thời gian, phù hợp cho các dự án lớn.

D. Free HTML template cho phép bạn tùy chỉnh CSS, thay đổi nội dung (văn bản, hình ảnh), tích hợp back-end (như Node.js, PHP).

🡪 C

Câu 11.3 Static files (tập tin tĩnh) trong Express là gì? Phát biểu nào sau đây không đúng?

A. Static files là các tập tin không thay đổi nội dung hoặc không cần xử lý logic phía server.

B. Static files được gửi trực tiếp cho client thông qua middleware express.static.

C. Mục đích của static files là tối ưu hóa hiệu suất, tổ chức dự án và bảo mật.

D. Không nên đặt các static files trong thư mục public.

🡪 D

**Web back-end (12) - Ứng dụng client-server**

Câu hỏi 12.1 Ứng dụng cục bộ là gì? Phát biểu nào sau đây không đúng?

A. Ứng dụng cục bộ chạy trực tiếp trên thiết bị của người dùng mà không cần kết nối liên tục với máy chủ ở xa.

B. 8754Ứng dụng cục bộ luôn yêu cầu kết nối internet ổn định để hoạt động hiệu quả.

C. Ứng dụng cục bộ có thể truy cập trực tiếp vào các tài nguyên của thiết bị như hệ thống tập tin và phần cứng.

D. Dữ liệu của ứng dụng cục bộ thường được lưu trữ trên thiết bị của người d/ùng, giúp tăng cường bảo mật và quyền riêng tư

🡪 B

Câu hỏi 12.2 Ứng dụng kiểu client-server là gì? Phát biểu nào sau đây không đúng?

A. Client tập trung chủ yếu vào việc hiển thị giao diện người dùng, trong khi server tập trung xử lý logic và dữ liệu.

B. Giao tiếp giữa client và server thường được thực hiện qua mạng, sử dụng các giao thức như HTTP, WebSocket hoặc TCP/IP.

C. Client và server luôn phải chạy trên các thiết bị riêng biệt, không bao giờ có thể cùng chạy trên một máy tính.

D. Server có nhiệm vụ nhận yêu cầu từ client, xử lý chúng, và gửi phản hồi lại cho client.

🡪 C

**Web back-end (13) - Lập trình client-server**

Bài tập 13.1 Viết đoạn mã thực hiện yêu cầu sau:

Từ trình duyệt, người dùng gửi GET request, có kèm theo MSSV, HoTen, DiemTrungBinh của một sinh viên.

Code trong index.js:

app.get('/', (req, res) => {

    const name = req.query.name;

    const mssv = req.query.mssv;

    const diemTB = req.query.diemTB;

    res.send(`

        <p>Chào bạn ${name}<p>

        <p>MSSV: ${mssv}</p>

        <p>Điểm trung bình của bạn là: ${diemTB}</p>

        <p>Chúc bạn học chăm!</p>

    `);

});

Gõ trên URL: <http://localhost:9000/?name=NguyenHoangSang&mssv=2212451&diemTB=8.0>

Câu hỏi 13.2 URL (Uniform Resource Locator) là gì? Phát biểu nào sau đây không đúng?

A. URL là một địa chỉ duy nhất xác định vị trí của một tài nguyên trên Internet.

B. URL bao gồm các thành phần như giao thức, tên miền, đường dẫn, truy vấn và phân mảnh.

C. Phần "truy vấn" (query) trong URL được sử dụng để truyền dữ liệu đến máy chủ thông qua các tham số.

D. Giao thức "FTP" là giao thức phổ biến nhất được sử dụng trong URL để truy cập các trang web.

🡪 D

**Web back-end (14) - Lập trình hướng sự kiện**

[1].Viết hàm theo kiểu thông thường:

const events = require('events'); 🡪 gọi Module event

const eventEmitter = new events.EventEmitter(); 🡪 tạo ra 1 eventEmitter

eventEmitter.on('hetGio', xuLyThongBao); 🡪 Đăng kí sự kiện,gắn với hàm lắng nghe

function xuLyThongBao(tb) { 🡪Xử lí sự kiện

    console.log(tb);

}

setTimeout(()=>{ 🡪 phát ra sự kiện, kèm xử lí ở trên

    eventEmitter.emit('hetGio', 'Hết giờ học rồi, về thôi!!!!!')

},2000);

[2].Viết hàm theo kiểu hàm mũi tên:

const events = require('events');

const eventEmitter = new events.EventEmitter();

eventEmitter.on('hetGio', (thongBao)=>{ 🡪 Khác biệt ở chỗ này

    console.log(thongBao);

});

setTimeout(()=>{

        eventEmitter.emit('hetGio', 'Hết giờ học rồi, về thôi!!!!!')

},2000);

Bài tập 14.2 Sử dụng module events của Nodejs để lập trình hướng sự kiện. Viết một chương trình đơn giản như sau:

- Khai báo sự kiện vaoLop (vào lớp), với hàm xử lý đi kèm là xuLyVaoLop

- Sử dụng hàm setTimeOut() để hẹn giờ vào lớp là 5 giây

- Hết 5 giây, web server sẽ thông báo “Đã đến giờ học” tới cửa sổ console của web server

- Viết bằng 2 cách: hàm thông thường và hàm mũi tên.

[1].Hàm mũi tên:

const events = require('events');

const eventEmitter = new events.EventEmitter();

eventEmitter.on('vaoLop', (thongBao)=>{

    console.log(thongBao);

});

setTimeout(()=>{

        eventEmitter.emit('vaoLop', 'Đã đến giờ học')

},5000);

[2].Hàm thông thường

const events = require('events');

const eventEmitter = new events.EventEmitter();

eventEmitter.on('vaoLop', xuLyVaoLop);

function xuLyVaoLop(tb) {

    console.log(tb);

}

setTimeout(()=>{

        eventEmitter.emit('vaoLop', 'Đã đến giờ học')

},5000);

Câu hỏi 14.3 Mô hình lập trình hướng sự kiện trong Node.js là gì? Phát biểu nào sau đây không đúng?

A. Event Emitter liên tục kiểm tra hàng đợi sự kiện và thực thi các hàm callback khi có sự kiện xảy ra.

B. Event là các hành động hoặc sự thay đổi trạng thái xảy ra trong ứng dụng.

C. Event Loop liên tục kiểm tra hàng đợi sự kiện và thực thi các hàm callback khi có sự kiện xảy ra.

D. Event Handler là các hàm được gọi khi một sự kiện xảy ra.

🡪A

**Web back-end (15) - Event-driven, Route handler và Middleware trong Express**

**– Web Server**

Câu hỏi 15.2 Mô hình lập trình hướng sự kiện của Express. Phát biểu nào sau đây không đúng?

A. Mỗi request từ client (như GET, POST, PUT, DELETE) được xem là một sự kiện (Event).

B. Các module http, fs, stream là các Bộ quản lý sự kiện (Event Emitter) của Express.

C. Express sử dụng Vòng lặp sự kiện (Event Loop) của trình duyệt để lắng nghe và xử lý các sự kiện theo kiểu bất đồng bộ (asynchronous), không chặn luồng (non-blocking).

D. Các route handler và middleware đóng vai trò như các Hàm xử lý sự kiện (Event Handler), được gọi khi sự kiện tương ứng xảy ra.

🡪 C

Câu hỏi 15.3 Route handler trong Express là gì? Phát biểu nào sau đây không đúng?

A. Route handler chỉ được gọi khi request khớp với route và HTTP method đã định nghĩa.

B. Route handler thường chứa logic nghiệp vụ như truy vấn cơ sở dữ liệu hoặc xử lý dữ liệu.

C. Route handler có thể chuyển tiếp request sang middleware hoặc handler tiếp theo bằng hàm next().

D. Chỉ có duy nhất một Route handler được gắn vào một route.

🡪 D

15.4. Middleware trong Express là gì? Phát biểu nào sau đây không đúng?

A. Middleware là các hàm được thực thi trong quá trình xử lý một HTTP request, trước hoặc sau Route handler.

B. Xử lý request (như xác thực, ghi log).

C. Sửa đổi đối tượng req hoặc res.

D. Middleware nhận 2 tham số: req, res.

🡪 D