

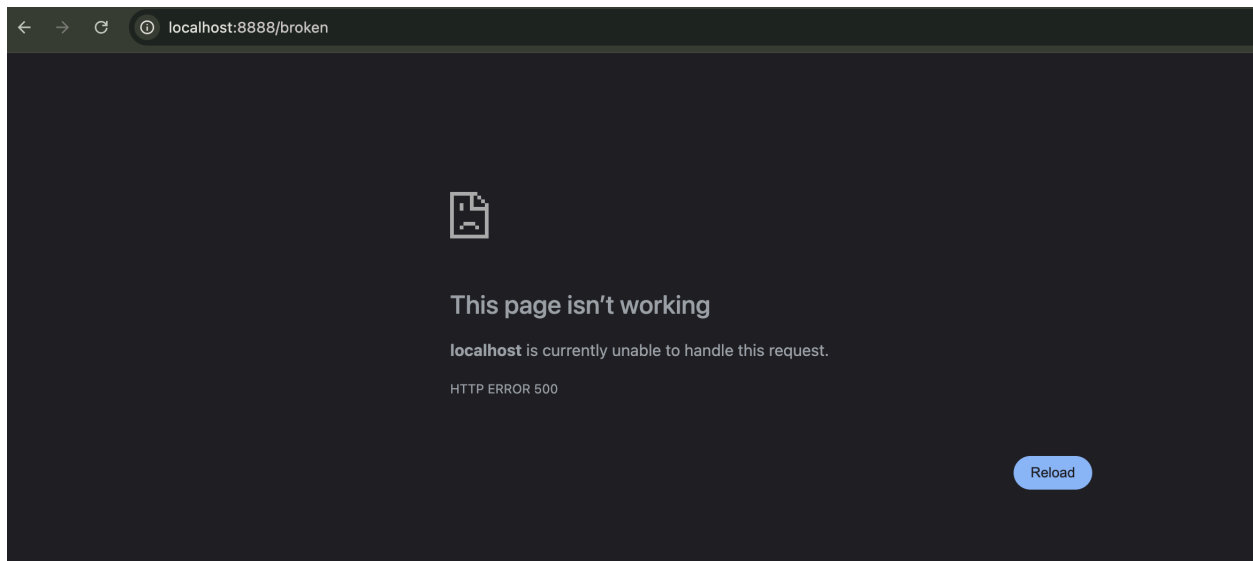
Pop Quiz

Challenge BackEnd:

First we run the project backend server



- Then we try to access the /broken route we get an 500 error as follows



- Basically we head back to the error-log file and check the log to understand the error log

```
~/Doc/V/S/pop-quiz cd /tmp ok | 33s | 13:49:41 ]
/tmp ls ok | 10:19:36 ]
com.apple.launchd.10D1loyWrj powerlog sprint1-php-error.log
/tmp cat sprint1-php-error.log ok | 10:19:37 ]
[11-Mar-2025 10:17:45 UTC] PHP Fatal error: Uncaught Error: Call to undefined method Slim\Psr7\Stream::write() in /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/index.php:41
Stack trace:
#0 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/Handlers/Strategies/RequestResponse.php(38): {closure}(Object(Slim\Psr7\Request), Object(Slim\Psr7\Response), Array)
#1 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/Routing/Route.php(363): Slim\Handlers\Strategies\RequestResponse->__invoke(Object(Closu
```

```

/tmp tail -f sprint1-php-error.log 1 err | 10:22:07
#2 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/MiddlewareDispatcher.php(73): Slim\Routing\Route->handle(Object(Slim\Psr7\Request))
#3 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/MiddlewareDispatcher.php(73): Slim\MiddlewareDispatcher->handle(Object(Slim\Psr7\Request))
#4 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/Routing/Route.php(321): Slim\MiddlewareDispatcher->handle(Object(Slim\Psr7\Request))
#5 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/Routing/RouteRunner.php(74): Slim\Routing\Route->run(Object(Slim\Psr7\Request))
#6 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/MiddlewareDispatcher.php(73): Slim\Routing\RouteRunner->handle(Object(Slim\Psr7\Request))
#7 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/App.php(209): Slim\MiddlewareDispatcher->handle(Object(Slim\Psr7\Request))
#8 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/vendor/slim/slim/Slim/App.php(193): Slim\App->handle(Object(Slim\Psr7\Request))
#9 /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/index.php(105): Slim\App->run()
#10 {main}
    thrown in /Users/void/Documents/VOID WORK/Sprint 1/sprint1-pop-quiz/backend/index.php on line 41

```

- So after checking the log we find that we have an error in line 41 in here:

```

39 $app->get('/broken', function (Request $request, Response $response, $args) {
40     /** @disregard P1013 because we're just testing */
41     $response->getBody()->write("Hello world!");
42     return $response;
43 });

```

- After taking further investigations: The issue in your `/broken` route is due to a **hidden Unicode character** in the method name `write`. Specifically, there's a **zero-width space (U+200B)** between `wr` and `ite`, which makes the method name invalid.

NOTE: We can clearly see the editor highlights it in yellow!

- After removing the zero-width space (U+200B), we fix the issue



A screenshot of a web browser window. The address bar shows 'localhost:8888/broken'. The page content displays 'Hello world!'.

- We run the command already explained previously

```

~/Doc/V/S/sprint1-pop-quiz | main !1 ab -n 200 -c 10 http://localhost:8888/crash
This is ApacheBench, Version 2.3 <$Revision: 1923142 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking localhost (be patient)
Completed 100 requests
Completed 200 requests
Finished 200 requests

Server Software:
Server Hostname: localhost
Server Port: 8888

Document Path: /crash
Document Length: 0 bytes

Concurrency Level: 10
Time taken for tests: 0.210 seconds
Complete requests: 200
Failed requests: 0
Non-2xx responses: 200
Total transferred: 70000 bytes
HTML transferred: 0 bytes
Requests per second: 954.62 [#/sec] (mean)
Time per request: 10.475 [ms] (mean)
Time per request: 1.048 [ms] (mean, across all concurrent requests)
Transfer rate: 326.29 [Kbytes/sec] received

```

- Again if we check the log file we fix the issue easily we identify the following:
 - The `count($logEntries)` function is being called incorrectly. The parentheses are misplaced, causing a syntax error.
 - Additionally, `0xA` is a hexadecimal value (equal to 10 in decimal), which might be confusing. It's better to use a decimal value for clarity.

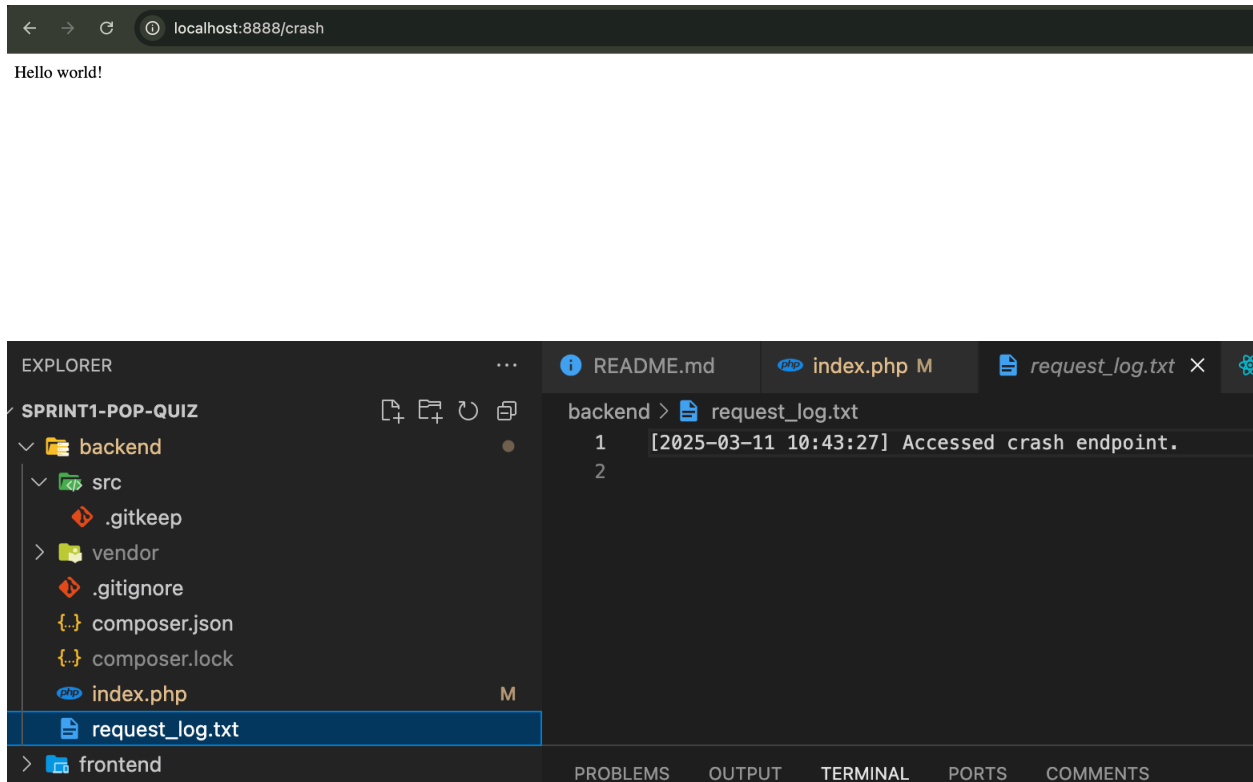
```

26     if (count($logEntries) > 0xA) {
27         // Unicode for line break \n
28         $contentClear = str_repeat('A', 0x9FFF0);
29         file_put_contents($logFile, $contentClear);
30     }
31     file_put_contents($logFile, $logEntry, FILE_APPEND);
32 }
33

```

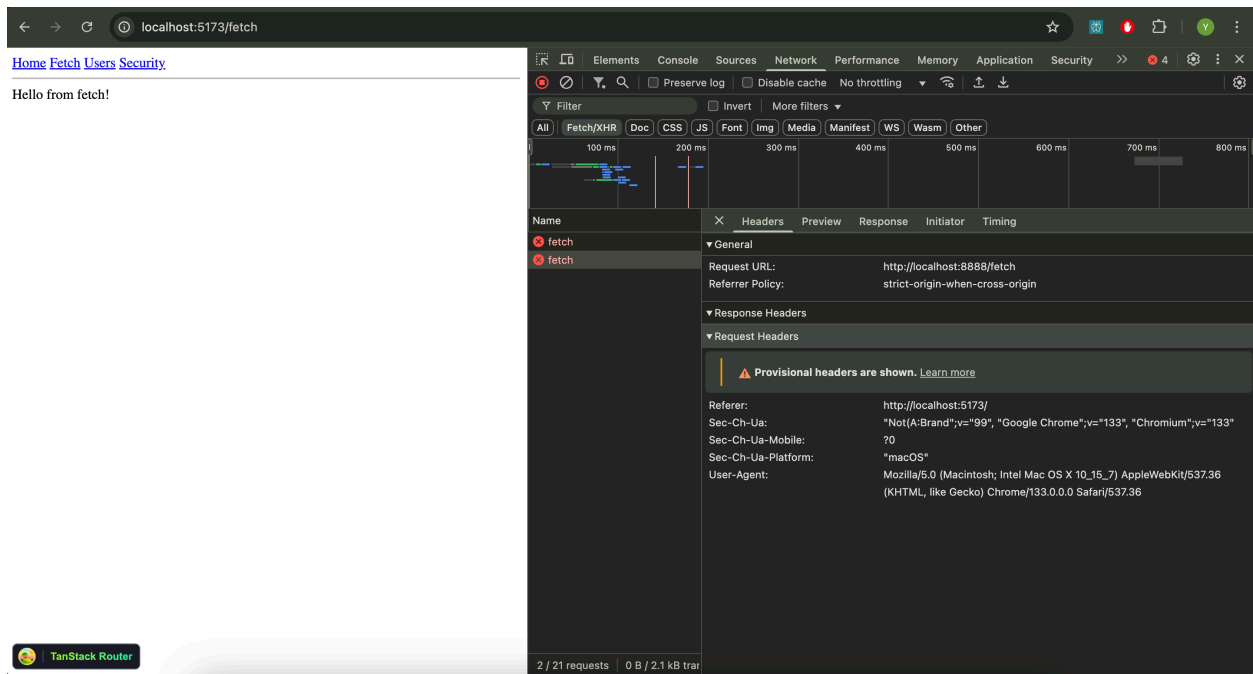
- We fix the function `logRequestWithRotation` to log request messages to a file (`request_log.txt`) while implementing a simple log rotation mechanism
 - Optional we can also fix the race condition that might encounter if not implemented the procedures to fix it.

→ Eventually the code works and we get request_log.txt log file created and handled correctly



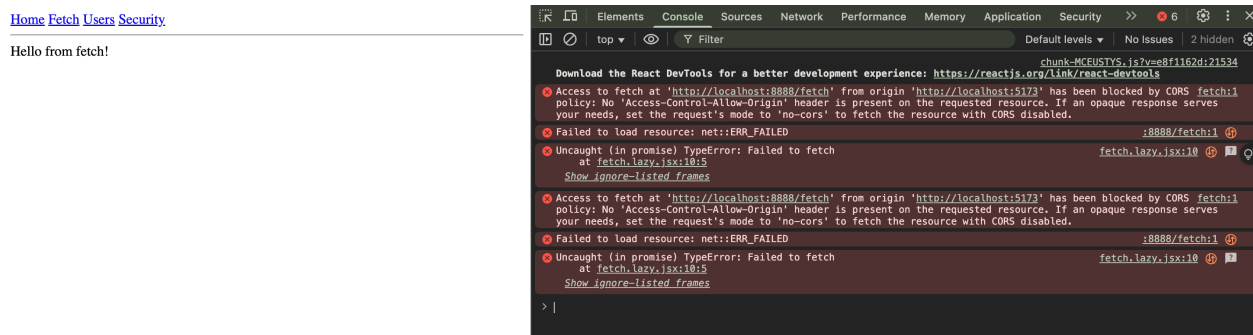
Challenge FrontEnd:

- This is the error we get when we fetch `/fetch`



The issue is that the app is trying to fetch data from a URL that is not accessible. The error message "Request headers are missing" suggests that the app is not sending the necessary headers to authenticate the request. This could be due to a misconfiguration in the app's code or a problem with the server's authentication system. To resolve this issue, the developer should check the app's code to ensure that the necessary headers are being sent and that the server's authentication system is functioning correctly.

- Here the error we get in more detail:



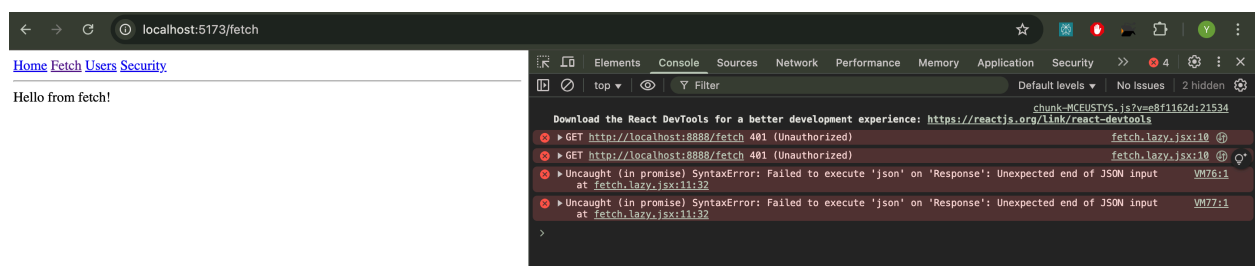
- **Cross-Origin:** Your JavaScript code (running in your browser at `http://localhost:5173`) is attempting to make a request to a different origin

(`http://localhost:8888/fetch`). Origins are considered different if they have different protocols (http/https), domains (localhost), or ports (5173/8888).

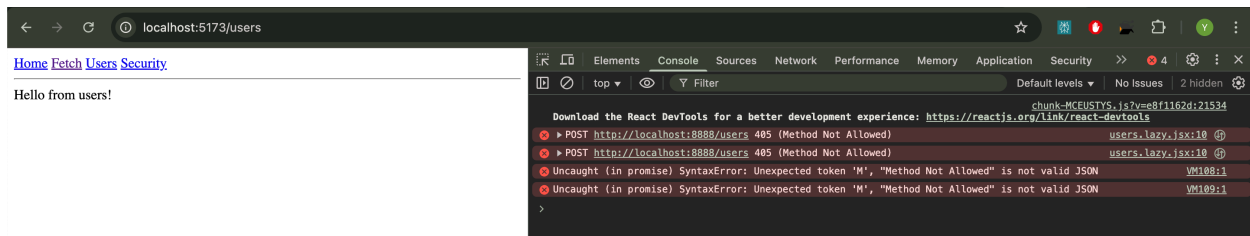
- **CORS Policy:** Browsers have a security feature called the Same-Origin Policy, which restricts web pages from making requests to different origins. CORS is a mechanism that allows servers to selectively relax this policy.
- **No 'Access-Control-Allow-Origin' Header:** The server at `http://localhost:8888` is not sending the `Access-Control-Allow-Origin` header in its response. This header is essential for CORS. It tells the browser whether the origin `http://localhost:5173` is allowed to access the resource.
- Now to fix we add our own header in the backend

```
You, 1 minute ago | 2 authors (Hamza Bahlouane and one other)
1 <?php
2 /* intelephense-disable */
3 session_start();
4 ini_set('memory_limit', '8M');
5 error_reporting(E_ALL);
6 ini_set('log_errors', 1);
7 ini_set('error_log', '/tmp/sprint1-php-error.log');
8 ini_set('display_errors', 0);
9
10
11 // Add CORS headers
12 header('Access-Control-Allow-Origin: http://localhost:5173'); // Replace with your React app's origin
13 header('Access-Control-Allow-Methods: GET, POST, OPTIONS'); // Allowed methods
14 header('Access-Control-Allow-Headers: Content-Type, Authorization'); // Allowed headers
15 header('Access-Control-Allow-Credentials: true'); // If you need to send cookies
16
17 if ($_SERVER['REQUEST_METHOD'] === 'OPTIONS') {
18     // Return early for preflight requests
19     http_response_code(204);
20     exit();
21 }
22 You, now • Uncommitted changes
```

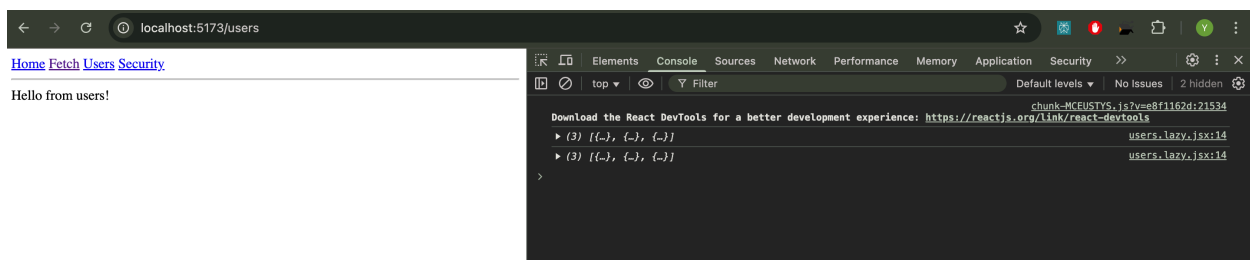
- as you can see we fixed the CORS error but we have another one to fix as well.



- Now in /users we get errr method not allowed



- We simply change the method from **POST** to **GET**



Optimization

- As for the optimization of loading we can use caching, just keep in mind browsers by default use caching by default or you can specify it and for how long you want the assets to be cached and so on. Here I'll share some methods you can use:

Cache-Control:

- **max-age=31536000** (1 year) for static assets that never change (images, fonts).
- **no-cache** for files that must always be revalidated with the server.
- **no-store** for files that should never be cached.

Expires:

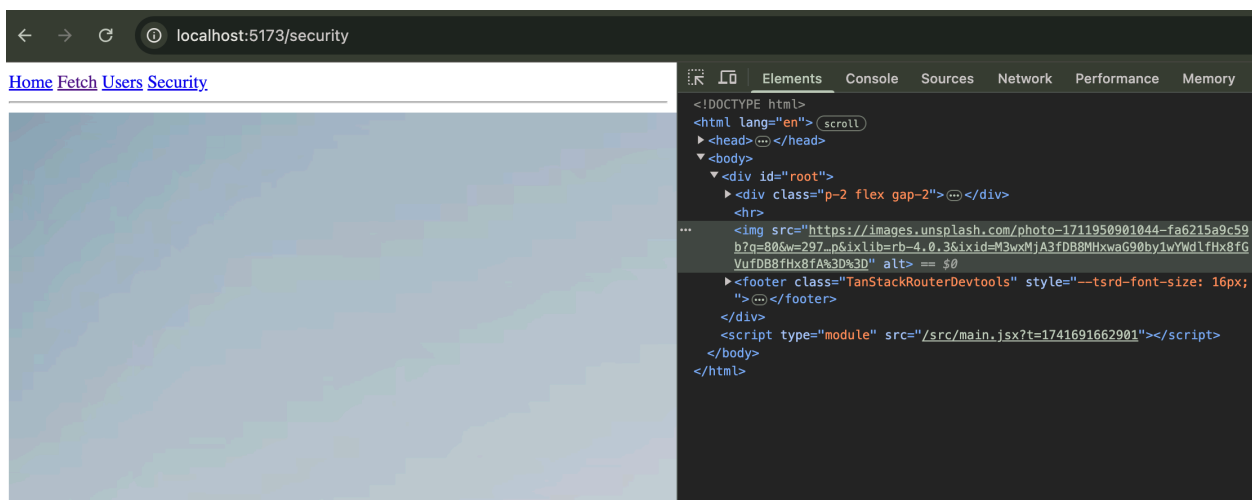
- A distant expiration date for static assets.

ETag and Last-Modified:

- These headers allow browsers to check if a resource has changed before downloading it again.

Security:

- The `/security` page loads an image from an untrusted source, allowing an attacker to inject malicious code through this image.
- For example, the image could contain JavaScript code that runs in the context of your website, allowing the attacker to steal sensitive data or modify the page content.



- How to fix this security issue use CSP

Attach the policy in the meta header of your page:

→ The Content Security Policy (CSP) is an added security layer that helps to detect and mitigate certain types of attacks, including cross-site scripting (XSS) and data injection attacks. CSP functions by allowing developers to control the resources that are allowed to load and execute on a web page. It helps to prevent attacks by specifying a whitelist of sources for different types of resources.

- I resolved it by adding the meta on the headers and appending it to the page that's why we see the error messages in the page

```
<meta http-equiv="Content-Security-Policy" content="default-src 'self' localhost; img-src 'self' localhost;">
```

localhost:5173/security

TANSTACK

React Router v1

Router8 items

state10 items

isLoading: false

isTransitioning: false

status: "pending"

resolvedLocation6 items

location6 items

matches2 items

pendingMatches: []

cachedMatches: []

statusCode: 200

redirect:

routesById5 items

routesByPath4 items

flatRoutes4 items

options6 items

defaultPreloadDelay: 50

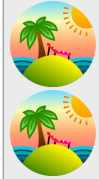
defaultPendingMs: 1000

defaultPendingMinMs: 500

context: {}

routeTree10 items

transformer: {}



ds: {}

ds: {}

rcTime

TanStack Router

Elements

Console

Sources

Network

Performance

Memory

Application

top

Filter

Default levels

5 issues

2 hidden

oG3CYr+VCSY5Zg1TKCqnu8IVlqlCjwJtclqG5mCIe'), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-ZCD5PKPLfOHGKvRyymu890Hnp0QxW/OSPT/VAJms='), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-5gcL/IUOKG8vCvA0dAhSDIK05Ijx5WjzII+QliJkyb0='), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-mPEb12zmiylwEupmY2L/SWuThXT+DYN0R0Hy2ELMosa'), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-a0nc1R8yF3K5Z0H0dHMLX1L1JGH8G0uPR6/ngVU='), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-5LXyK7K5sJcXJdLJA06IY+LeYF6pSugMOL+aeQKc='), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-Gk1rW45IAWu9d9KX9G081la+9j+CF6zwxCCiy9NSka'), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-6VL5G55zqMwyDfK9kxwLU5Prq5SRGinTaRg0xy8CU='), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

» Refused to apply inline style because it violates the following Content Security Policy directive: "default-src 'self' localhost". Either the 'unsafe-inline' keyword, a hash ('sha256-r1ePlJuvIAxi1H7PK14z0j005FZFioR4S6BQayz1fE='), or a nonce ('nonce-...') is required to enable inline execution. Note also that 'style-src' was not explicitly set, so 'default-src' is used as a fallback.

Pop Quiz

10