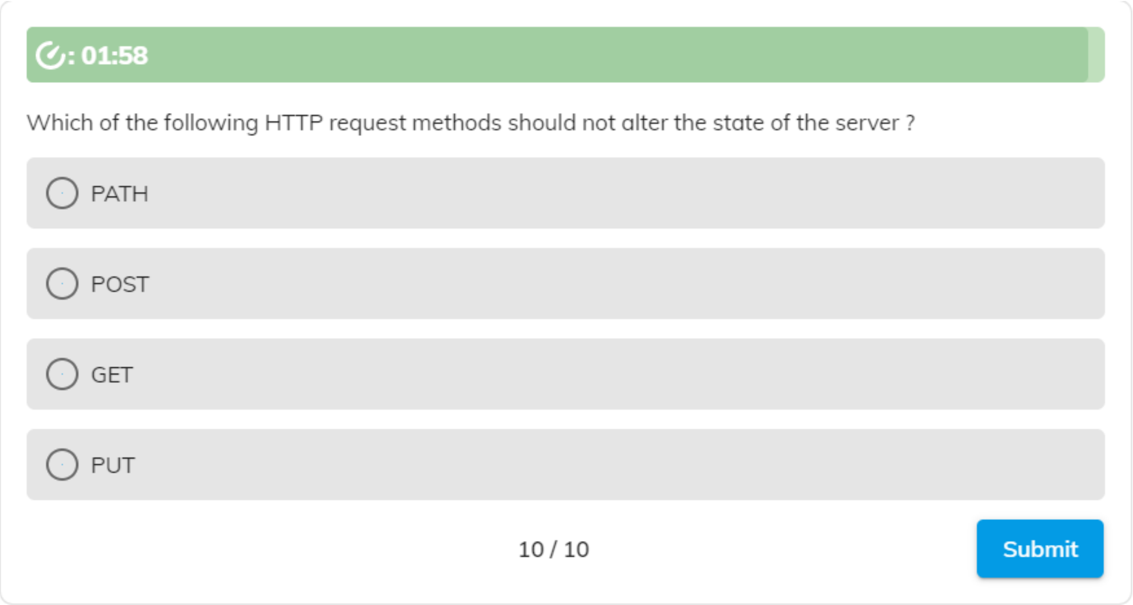


Which of the following HTTP request methods should not alter the state of the server ?

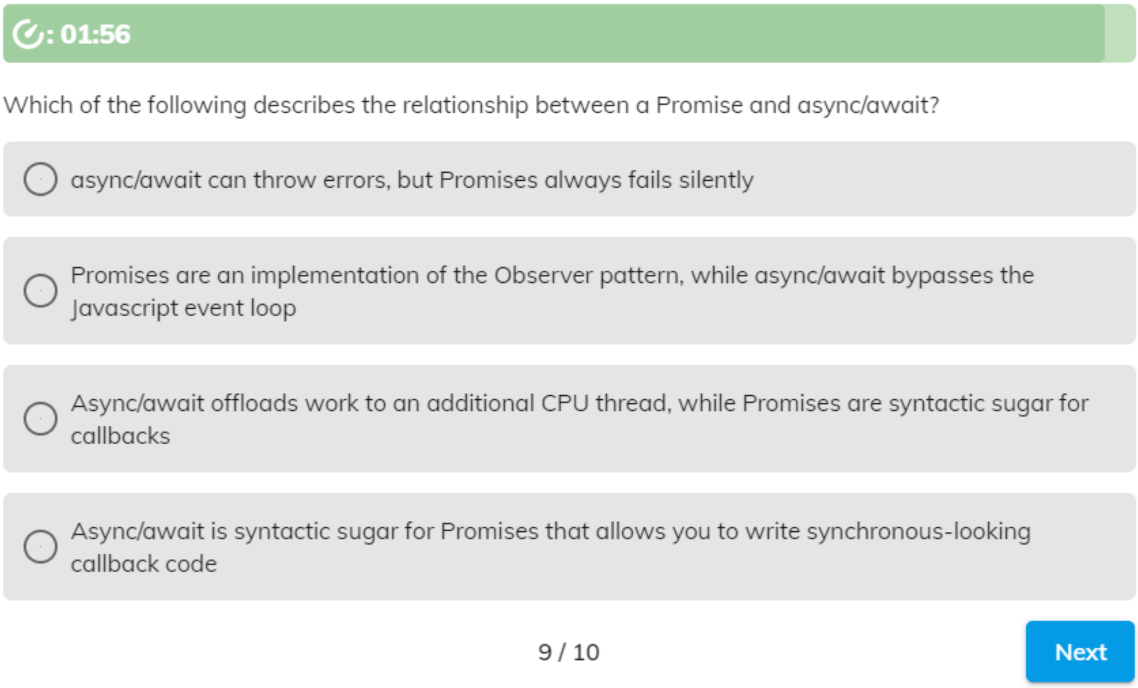
○ PATH

○ POST

○ GET

○ PUT

10 / 10



Which of the following describes the relationship between a Promise and async/await?

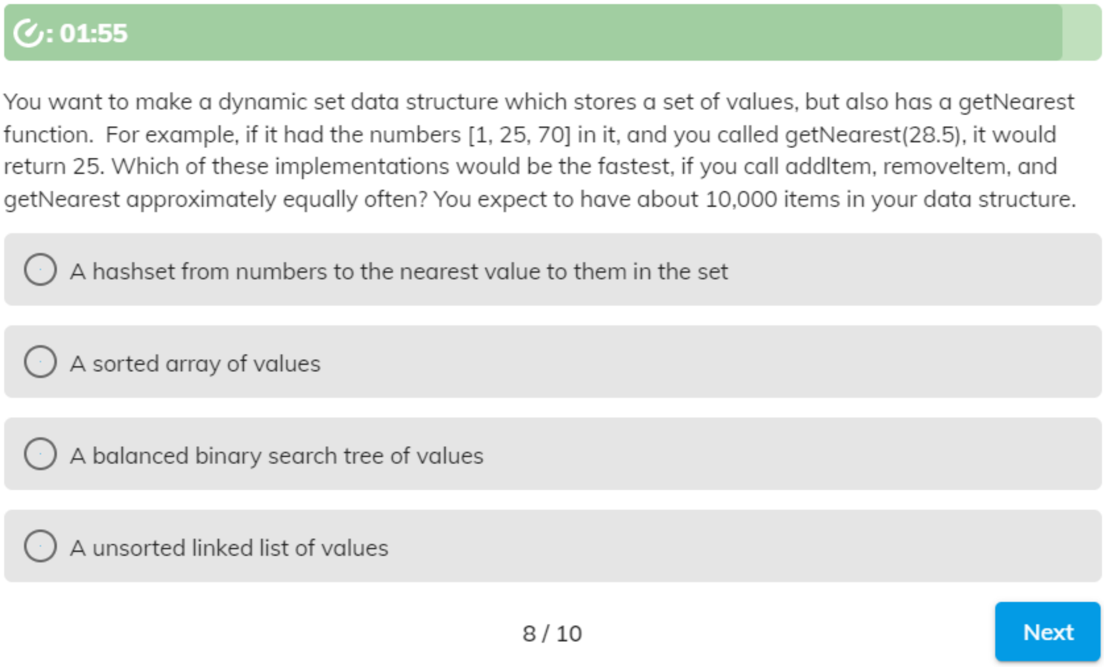
○ async/await can throw errors, but Promises always fails silently

○ Promises are an implementation of the Observer pattern, while async/await bypasses the Javascript event loop

○ Async/await offloads work to an additional CPU thread, while Promises are syntactic sugar for callbacks

○ Async/await is syntactic sugar for Promises that allows you to write synchronous-looking callback code

9 / 10



You want to make a dynamic set data structure which stores a set of values, but also has a getNearest function. For example, if it had the numbers [1, 25, 70] in it, and you called getNearest(28.5), it would return 25. Which of these implementations would be the fastest, if you call addItem, removeItem, and getNearest approximately equally often? You expect to have about 10,000 items in your data structure.

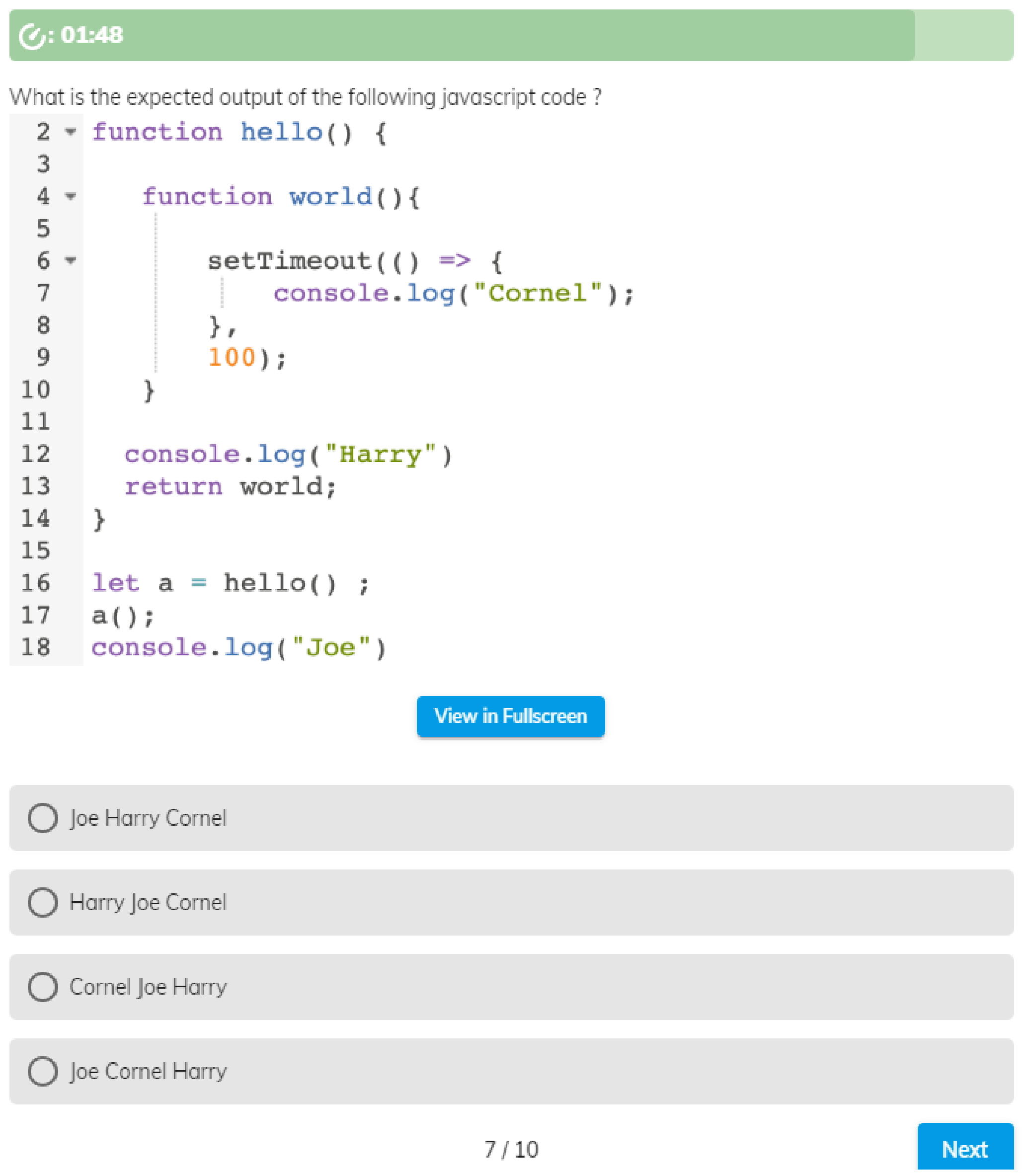
○ A hashset from numbers to the nearest value to them in the set

○ A sorted array of values

○ A balanced binary search tree of values

○ A unsorted linked list of values

8 / 10



What is the expected output of the following javascript code ?

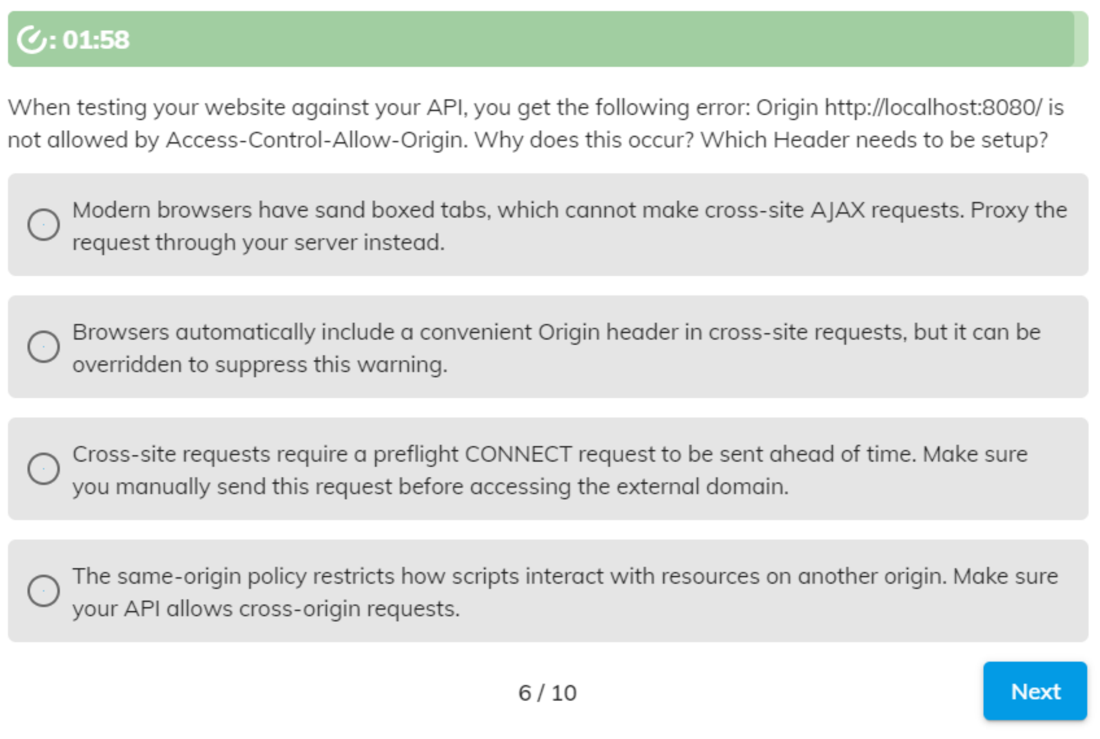
```javascript
function hello() {

   function world(){

       setTimeout(() => {
           console.log("Cornel");
       },
       100);
   }

  console.log("Harry")
  return world;
}

let a = hello() ;
a();
console.log("Joe")
```

○ Joe Harry Cornel

○ Harry Joe Cornel

○ Cornel Joe Harry

○ Joe Cornel Harry



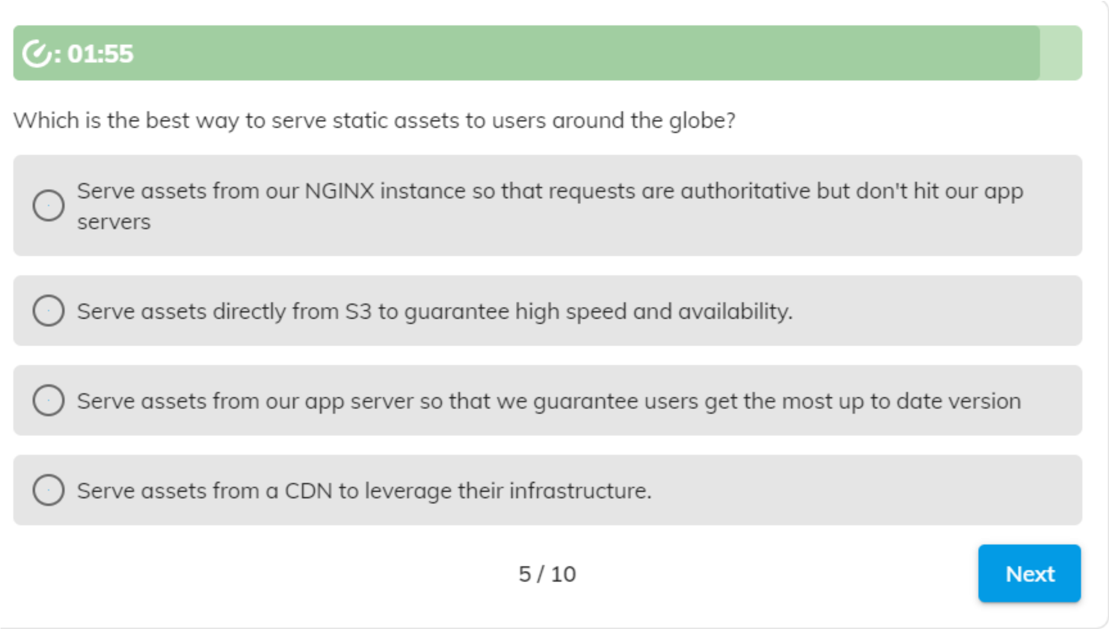
When testing your website against your API, you get the following error: Origin http://localhost:8080/ is not allowed by Access-Control-Allow-Origin. Why does this occur? Which Header needs to be setup?

○ Modern browsers have sand boxed tabs, which cannot make cross-site AJAX requests. Proxy the request through your server instead.

○ Browsers automatically include a convenient Origin header in cross-site requests, but it can be overridden to suppress this warning.

○ Cross-site requests require a preflight CONNECT request to be sent ahead of time. Make sure you manually send this request before accessing the external domain.

○ The same-origin policy restricts how scripts interact with resources on another origin. Make sure your API allows cross-origin requests.



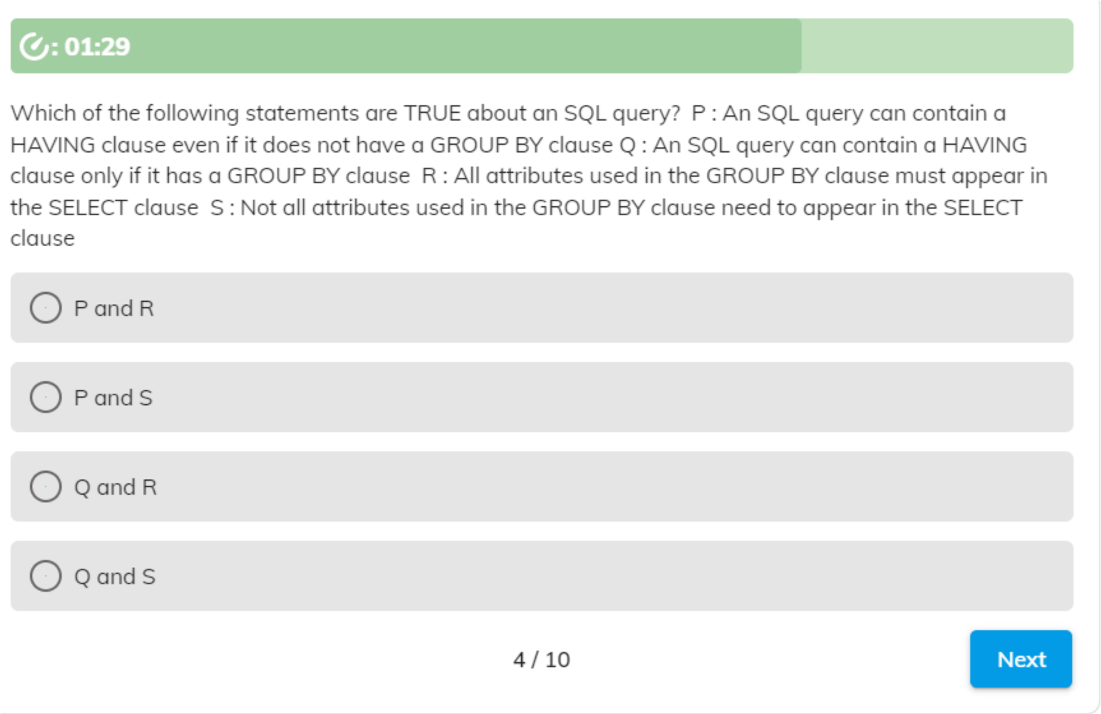
Which is the best way to serve static assets to users around the globe?

○ Serve assets from our NGINX instance so that requests are authoritative but don't hit our app servers

○ Serve assets directly from S3 to guarantee high speed and availability.

○ Serve assets from our app server so that we guarantee users get the most up to date version

○ Serve assets from a CDN to leverage their infrastructure.



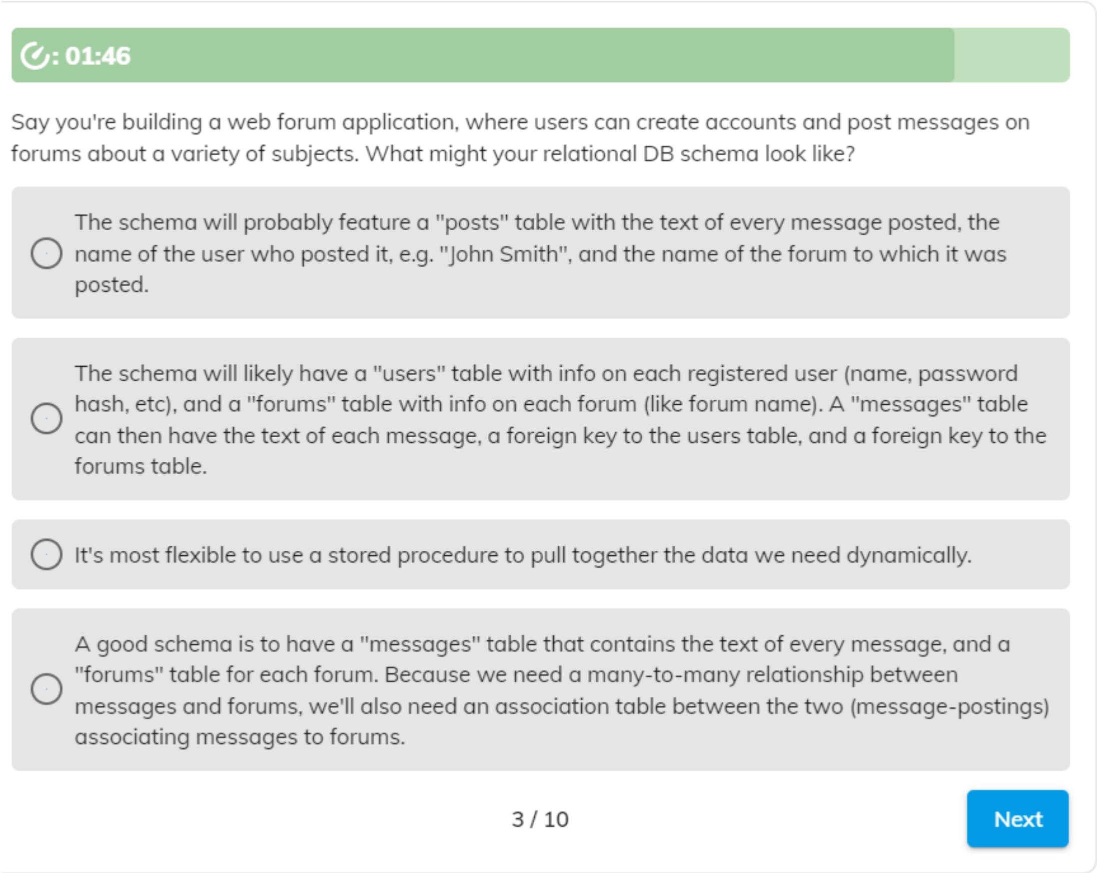
Which of the following statements are TRUE about an SQL query? P : An SQL query can contain a HAVING clause even if it does not have a GROUP BY clause Q : An SQL query can contain a HAVING clause only if it has a GROUP BY clause R : All attributes used in the GROUP BY clause must appear in the SELECT clause S : Not all attributes used in the GROUP BY clause need to appear in the SELECT clause

○ P and R

○ P and S

○ Q and R

○ Q and S



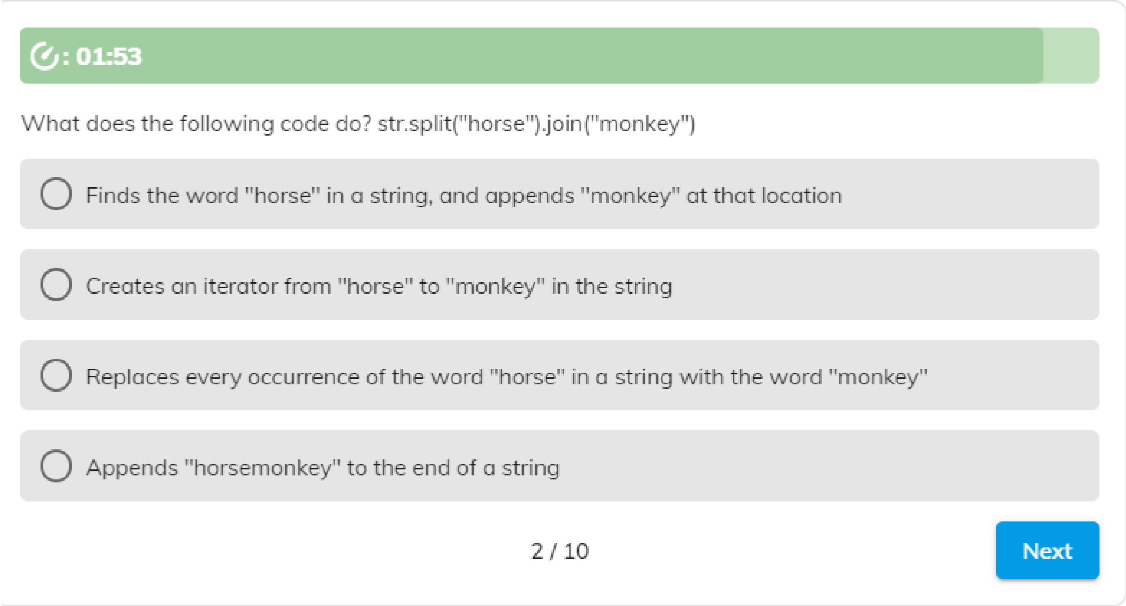
Say you're building a web forum application, where users can create accounts and post messages on forums about a variety of subjects. What might your relational DB schema look like?

○ The schema will probably feature a "posts" table with the text of every message posted, the name of the user who posted it, e.g. "John Smith", and the name of the forum to which it was posted.

○ The schema will likely have a "users" table with info on each registered user (name, password hash, etc), and a "forums" table with info on each forum (like forum name). A "messages" table can then have the text of each message, a foreign key to the users table, and a foreign key to the forums table.

○ It's most flexible to use a stored procedure to pull together the data we need dynamically.

○ A good schema is to have a "messages" table that contains the text of every message, and a "forums" table for each forum. Because we need a many-to-many relationship between messages and forums, we'll also need an association table between the two (message-postings) associating messages to forums.



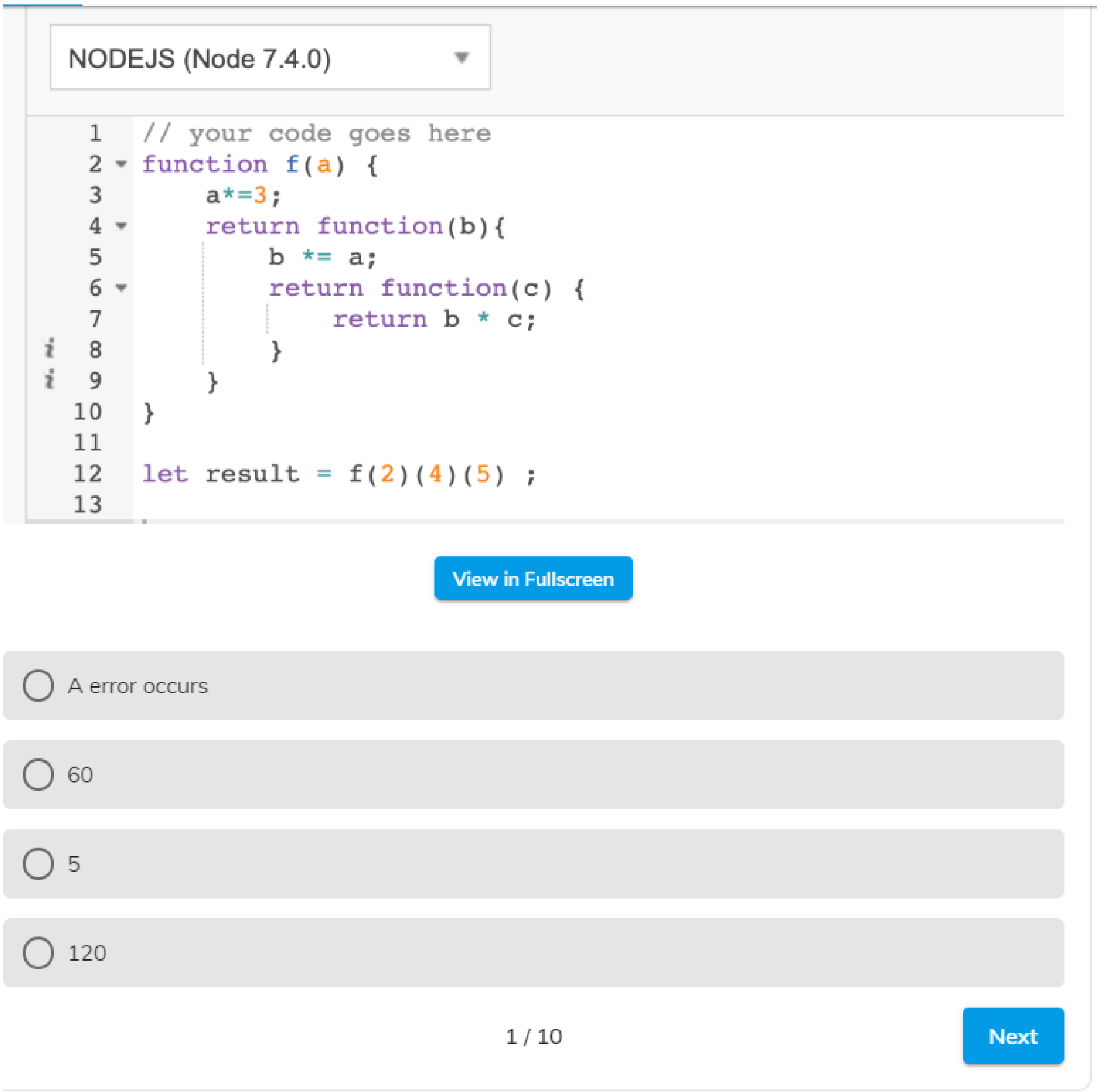
What does the following code do? str.split("horse").join("monkey")

○ Finds the word "horse" in a string, and appends "monkey" at that location

○ Creates an iterator from "horse" to "monkey" in the string

○ Replaces every occurrence of the word "horse" in a string with the word "monkey"

○ Appends "horsemonkey" to the end of a string



NODEJS (Node 7.4.0)

```javascript
// your code goes here
function f(a) {
    a*=3;
    return function(b){
        b *= a;
        return function(c) {
            return b * c;
        }
    }
}

let result = f(2)(4)(5) ;

```

○ A error occurs

○ 60

○ 5

○ 120

1 / 10