## Questions

### Biography:

How old are you?

* 24

Have you used JavaScript before?

* A little bit of experience

Have you ever written asynchronous code? (async, await, promise, then, …)

* Llittle experience with it

From a scale of 1-10 rate your understanding of how asynchronous code works in JavaScript?

* 3/10

### Reading the code:

***Now pass the code example to the user. Give them enough time to read and understand the code.***

Did you fully understand the functionality of the provided code?

* Yes, understands relatively what the program is trying to do

Describe the control flow of the code using the line numbers

* Plan trip is called in line 37
* Goes to line 21 where the function planTrip is called
* Line 22 calls line 1
* Go back to 23 and logs destinationDetails
* If statement checks if destination is unavailable in lines 25 to 27 and return empty if its unavailable
* Then calls lines 29-31 and gets all the details for hotels, flights, and cars
  + 29 goes to 6
  + 30 goes to 11
  + 31 goes to 16
* When it returns it will go back to line 38 and then log the results in line 39

What would the overall runtime be? (Give a minimum in ms)

* It’ll take around 2500 ms

Can you think of any changes you can make to the code that will reduce the overall runtime?

* Fetch hotel, flights, cars all at the same time

### Our Program:

***Now show the expected results of the example that our program produces to the user.***

Were there any changes that the user did not come up with that the program did?

* Didn’t notice the if-statement could be moved before the first fetch

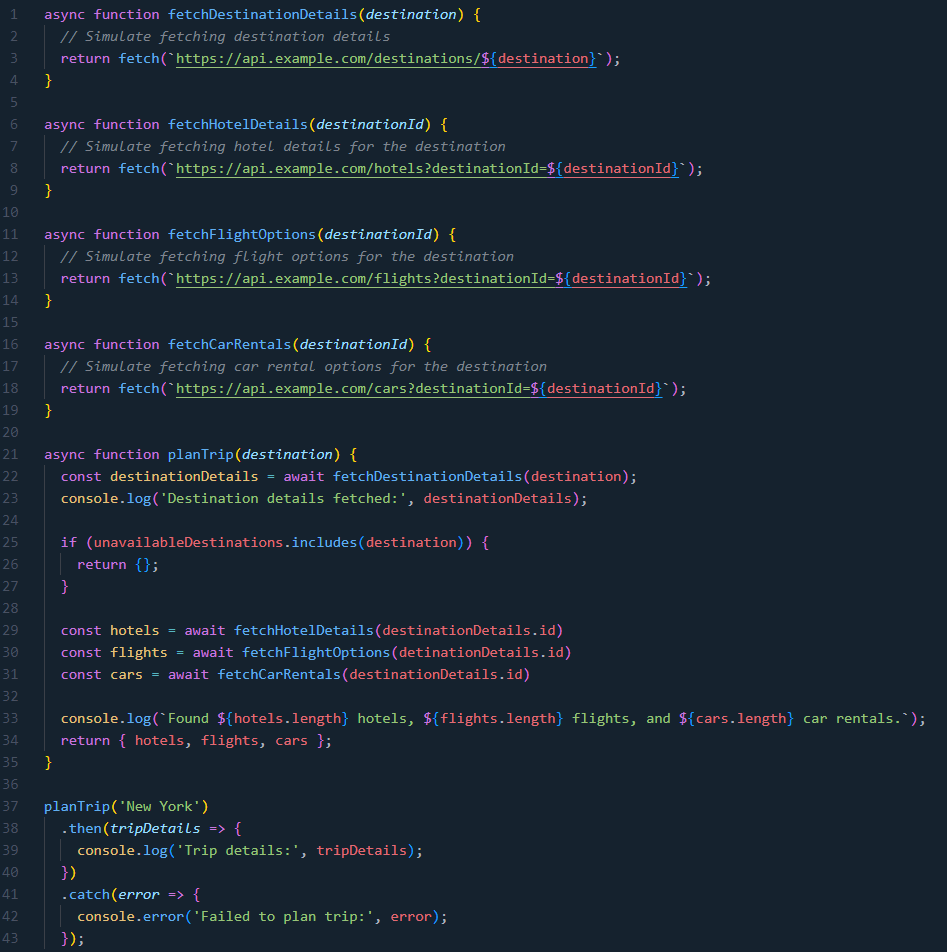
Were there any changes that the program did not come up with but the user did?

* No

Is the program output easy to understand? If not, provide any feedback

* Yes it’s easy to understand, just make sure to print the feedback clearly
  + Feedback should be similar to what chatGPT gives you
  + Suggestions with examples?
  + In this case, it’d suggest using Promise.all but not necessarily the full piece of optimized code

## Code



Assume that fetchDestinationDetails takes 1000ms, fetchhotelDetails takes 500ms, fetchFlightOptions takes 400ms, and fetchCarRentals takes 600ms.

What our program would suggest doing:



In the inefficient example the total runtime would be at least (1000ms + 500ms + 400ms + 600ms) = 2500ms as we’re fetching hotel details, flight options, and car rentals one at a time. By using Promise.all we’re taking advantage of fetching the data in parallel reducing the total runtime to at least (1000ms + max(500ms, 400ms, 600ms)) = 1600ms. The program would also suggest to move the if statement in planTrip to the beginning as it’s blocked from running as it waits for fetchDestinationDetails even though it doesn’t need the results from that async call to do the check.