Short Answer:

Answer the following questions with complete sentences in **your own words**. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers.

1. What are the disadvantages of synchronous code?
2. What is asynchronous code in JavaScript?
3. Since JS is single-threaded, how does it achieve asynchronous code?
4. What does the event loop do? What data structures does it use?
5. What is the callback queue?
6. What is an HTTP request and HTTP response?
7. How many HTTP methods are there? Explain each one.
    1. What is the difference between GET and POST? What about POST and PUT?
8. Could you explain the different classes of HTTP status codes? What are some common status codes?
9. What is AJAX?
10. What is XHR?
11. What is a Promise?
12. How many states does a Promise have? What are they?
13. What is callback hell?
14. What is the advantage of Promises over callbacks?
15. Explain Promise.all() vs Promise.allSettled().
16. What is the Microtask Queue?
17. What is the difference between making server requests via fetch and XHR?
18. What is async & await? How do we use them?
19. Explain localStorage vs sessionStorage.
20. What do the async and defer attributes do when loading scripts?
21. What are ES6 modules? Why are they useful? How do we make these modules available in other JS files?

Coding Questions:

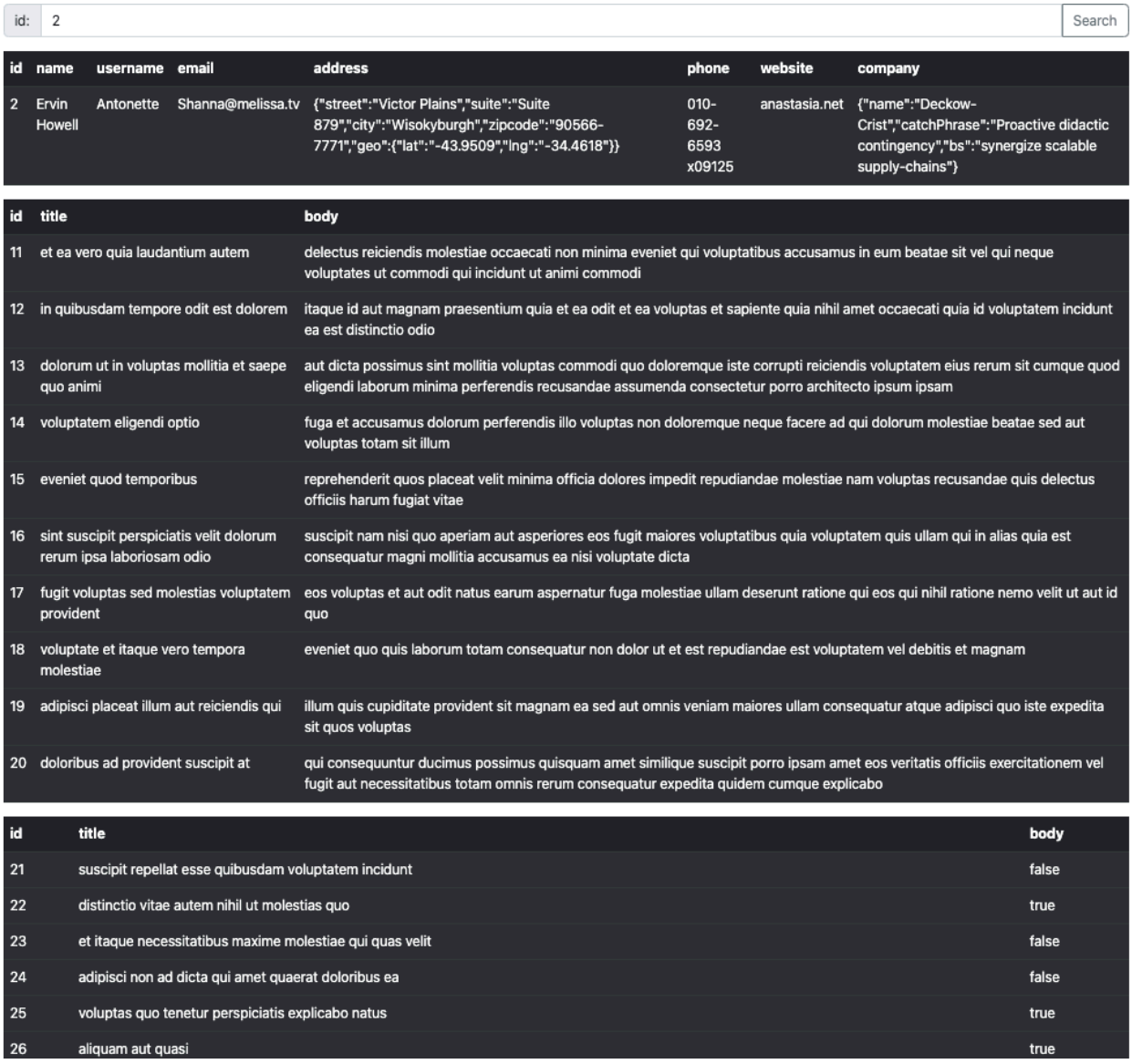
Use HTML/CSS/JS to solve the following problems. You are highly encouraged to present more than one way to answer the questions. Please follow best practices when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

1. Given a url **“https://jsonplaceholder.typicode.com/users"**, send a GET request to display **all** the data on the page in a table. You may use **JSON.stringify()** to display the properties with nested objects. Errors should be handled properly.
   - Do this with fetch and XHR.

2. Create a webpage with text input and a search button. When you input a user ID and click search, it should display all of that user’s information, posts, and todos at the same time on the same page in a table. **Hint: Promise.all() or Promise.allSettled().**

- For example, when the user types 2, display the data from the following urls:
  https://jsonplaceholder.typicode.com/users/2
  https://jsonplaceholder.typicode.com/posts?userId=2
  https://jsonplaceholder.typicode.com/todos?userId=2

- If the user ID is invalid (no data in the response), there should be an error message says “User was not found. Please try another user ID”.



id: 2 Search

id	name	username	email	address	phone	website	company
2	Ervin Howell	Antonette	Shanna@melissa.tv	{"street":"Victor Plains","suite":"Suite 879","city":"Wisokyburgh","zipcode":"90566-7771","geo":{"lat":"-43.9509","lng":"-34.4618"}}	010-692-6593 x09125	anastasia.net	{"name":"Deckow-Crist","catchPhrase":"Proactive didactic contingency","bs":"synergize scalable supply-chains"}

id	title	body
11	et ea vero quia laudantium autem	delectus reiciendis molestiae occaecati non minima eveniet qui voluptatibus accusamus in eum beatae sit vel qui neque voluptates ut commodi qui incidunt ut animi commodi
12	in quibusdam tempore odit est dolorem	itaque id aut magnam praesentium quia et ea odit et ea voluptas et sapiente quia nihil amet occaecati quia id voluptatem incidunt ea est distinctio odio
13	dolorum ut in voluptas mollitia et saepe quo animi	aut dicta possimus sint mollitia voluptas commodi quo doloremque iste corrupti reiciendis voluptatem eius rerum sit cumque quod eligendi laborum minima perferendis recusandae assumenda consectetur porro architecto ipsum ipsam
14	voluptatem eligendi optio	fuga et accusamus dolorum perferendis illo voluptas non doloremque neque facere ad qui dolorum molestiae beatae sed aut voluptas totam sit illum
15	eveniet quod temporibus	reprehenderit quos placeat velit minima officia dolores impedit repudiandae molestiae nam voluptas recusandae quis delectus officiis harum fugiat vitae
16	sint suscipit perspiciatis velit dolorum rerum ipsa laboriosam odio	suscipit nam nisi quo aperiam aut asperiores eos fugit maiores voluptatibus quia voluptatem quis ullam qui in alias quia est consequatur magni mollitia accusamus ea nisi voluptate dicta
17	fugit voluptas sed molestias voluptatem provident	eos voluptas et aut odit natus earum aspernatur fuga molestiae ullam deserunt ratione qui eos qui nihil ratione nemo velit ut aut id quo
18	voluptate et itaque vero tempora molestiae	eveniet quo quis laborum totam consequatur non dolor ut et est repudiandae est voluptatem vel debitis et magnam
19	adipisci placeat illum aut reiciendis qui	illum quis cupiditate provident sit magnam ea sed aut omnis veniam maiores ullam consequatur atque adipisci quo iste expedita sit quos voluptas
20	doloribus ad provident suscipit at	qui consequuntur ducimus possimus quisquam amet similique suscipit porro ipsam amet eos veritatis officiis exercitationem vel fugit aut necessitatibus totam omnis rerum consequatur expedita quidem cumque explicabo

id	title	body
21	suscipit repellat esse quibusdam voluptatem incidunt	false
22	distinctio vitae autem nihil ut molestias quo	true
23	et itaque necessitatibus maxime molestiae qui quas velit	false
24	adipisci non ad dicta qui amet quaerat doloribus ea	false
25	voluptas quo tenetur perspiciatis explicabo natus	true
26	aliquam aut quasi	true

3. Implement a function **delayedRequest(url)** that retrieves data from the specified url and outputs it to the console after 2 seconds.
- Test it with any of the “https://jsonplaceholder.typicode.com/users" urls.

Paired Programming:

Use JS to solve the following problems with your partner. Please remember to label who was acting as the driver and navigator for each problem and write down your feedback on their performance. **Feedback will be confidential**.

Leetcode #66: Plus One
Leetcode #70: Climbing Stairs