HW5 - Javascript 3

0. Leetcode

https://leetcode.com/problem-list/oizxjoit/

每天1-3道题，需要用JavaScript或者TypeScript

1. 问答练习(八股)

准备以下八股题目答案, 写在`note.md`里

1. What is the difference between innerHTML and textContent?
2. What are the differences between call, apply & bind?
3. Explain the this keyword in JavaScript.
4. What does the event loop do? What data structures does it use?
5. What is the callback queue?
6. What are closures?
7. What is asynchronous code in JavaScript? How does JavaScript achieve asynchronous code?
8. What is async & await? How do we use them?
9. How many HTTP methods are there? Explain each one.
    a. What is the difference between GET and POST? What about POST and PUT?
10. What is a Promise?
11. What is promise chaining?
12. Explain the three states of a Promise.
13. What is the use of Promise.all()? How is it different from Promise.allSettled?
14. What is the advantage of Promises over callbacks?
15. Describe the difference between a cookie, sessionStorage and localStorage in browsers.

小组间Peer Mock，录音并上传

2. Coding

1. Given the sample UI, implement the following product management page with styling that
matches as closely as possible.

a. The user can fill out the fields and click "Add New" to create a new product in the table. However, if any of the input fields are empty, no product should be created.

b. After the user create a new product in the table successfully, the input fields should be cleared.

c. The user can delete existing products in the table by clicking the delete button.

d. By default, when the user loads the page for the first time, there should be these 3 items in the table as shown below.

e. Style the even rows with a darker background.

Product Name	Product Category	Product Price	Action
M&M	Snacks	$1.99	Delete
Table	Furniture	$199	Delete
Kale	Vegetables	$2.49	Delete

Add Product

Product Name: Product Category: Product Price: Add New

2. Use HTML/CSS/JS to solve the following problems. Please follow best practices when you write the code so that it would be easily readable, maintainable, and efficient.

   a. [Part 1] Given a url `https://jsonplaceholder.typicode.com/users`, send a GET request to display the data on the page in a **table**. Errors should be handled properly.

   b. [Part 2] Create a text input box and a search button. When you input a user ID and click search, it should display that user's information, posts, and todos all in the same page in a **list** with the format of key: value. *(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

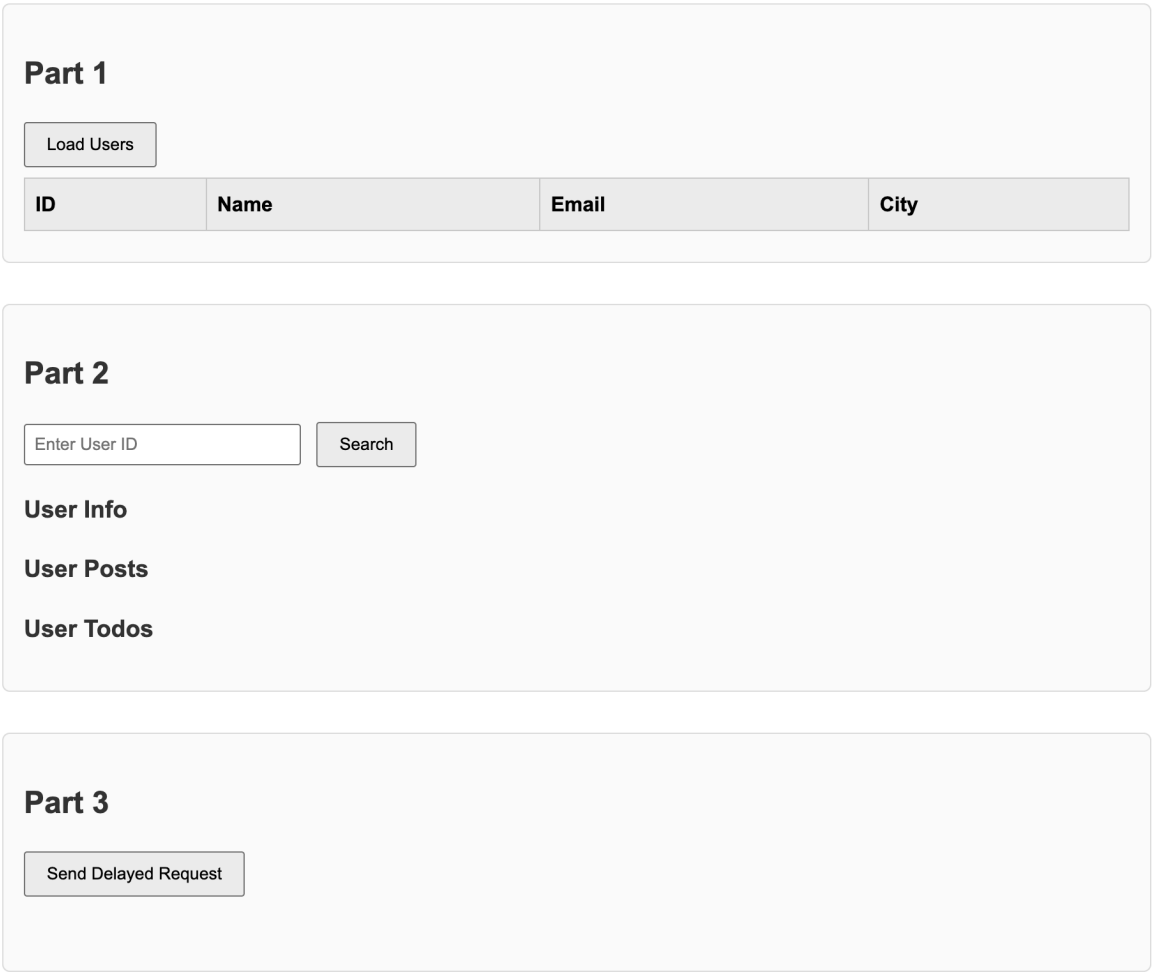
https://jsonplaceholder.typicode.com/todos?userId=2

- 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` and then clear the input box.

c. [Part 3] Implement a function **delayedRequest(url)** that fetches data from the url and outputs the json string data to the **console** after 2 seconds. *(Hint: JSON.stringify(data, null, " "))*

- After clicking the button, immediately display `Waiting ...` on the page.
- After the data is retrieved, replace the message with `Check console for the data` in the same area.
- Test it with any of the “https://jsonplaceholder.typicode.com/users/${id}" urls. You may set a default value for the function parameter.

Put the three parts on a single page, similar to the example below:



Sample web page