ITC5202: Project 2 Report

Created by: <Xinfu Guo>

Student ID: <N01611988>

Date: <2024/4/14>

# Table of contents

[Table of contents 1](#_Toc360911754)

[Task 1 2](#_Toc693012242)

[Task 2 3](#_Toc157504488)

[Task 3 and 4 4](#_Toc280152798)

[Task 5 5](#_Toc835990601)

[Task 6 6](#_Toc1362278575)

[Task 7 7](#_Toc396065624)

[Task 8 8](#_Toc974513945)

[Task 9 9](#_Toc1437716659)

[Task 10 10](#_Toc1528208752)

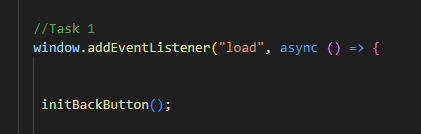
[Task 11 11](#_Toc14270007)

[Summary 12](#_Toc1409141569)

## Task 1

Describe the purpose of the load event handler.

It will be runed when this page is loading.



## Task 2

Use console.log to log the parsed JSON to the browser console. Take a screenshot of the console

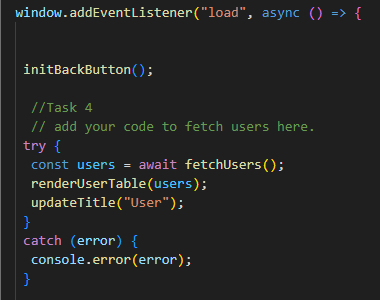


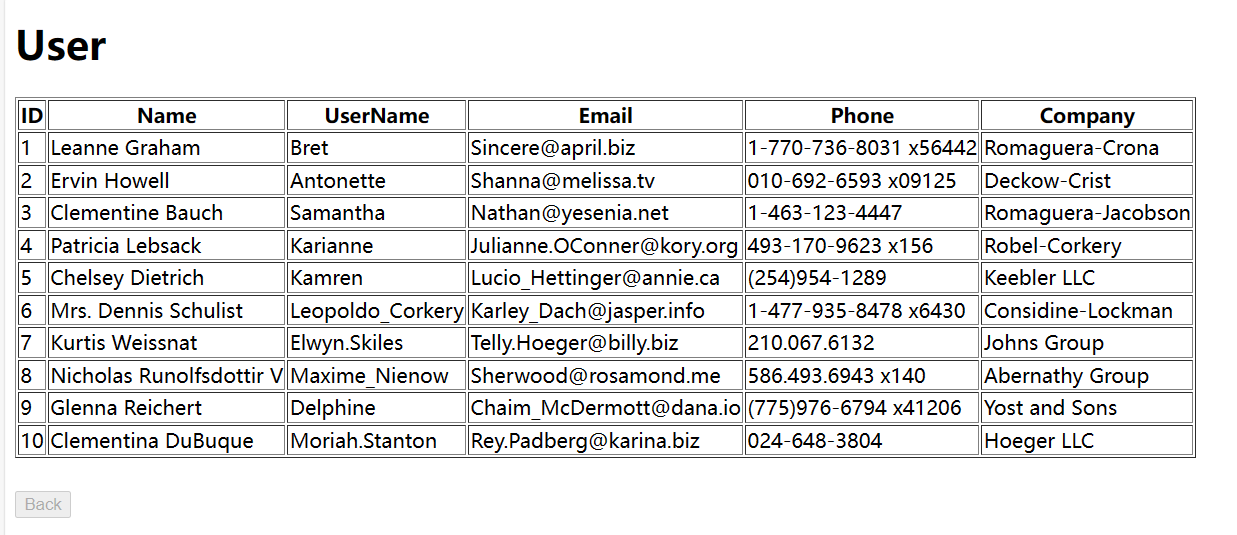


## Task 3 and 4

Take a screenshot of the rendered html table. Describe how you would “catch” any errors from your function.

In the try block, I can throw new exception, and also when the code is running, the exception happened in the try block will be caught by catch block, then run the code in catch block, which is a elegant way to solve exception





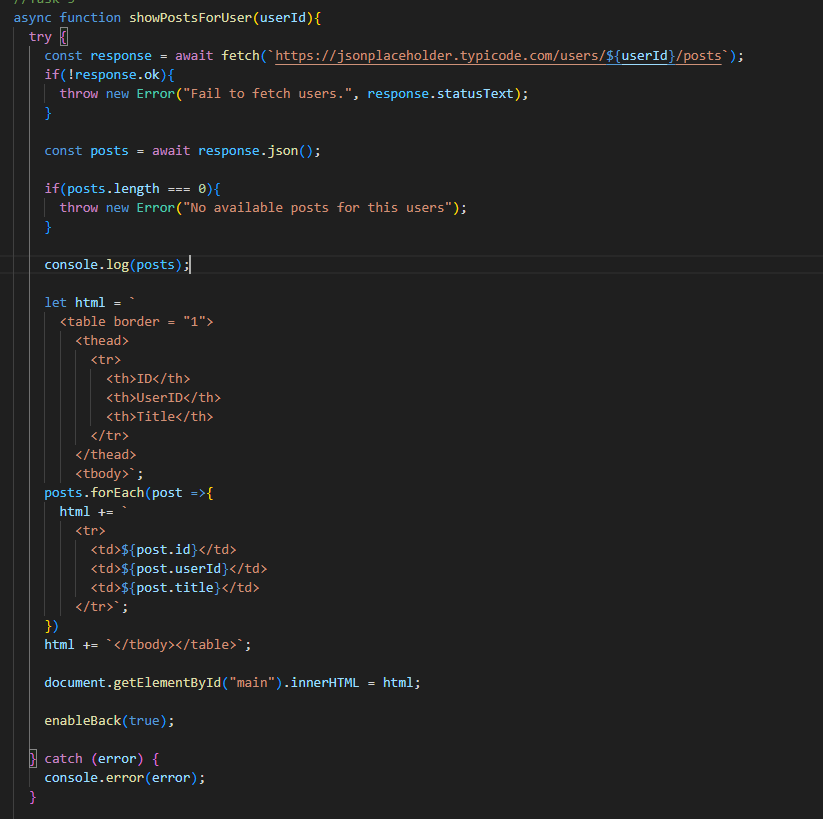
## Task 5

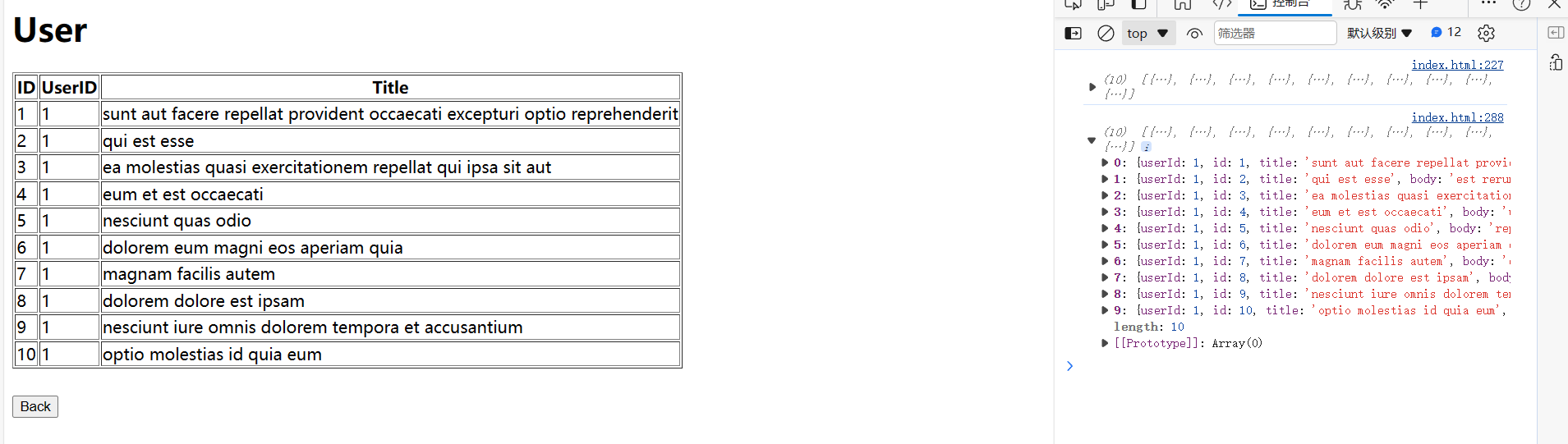
Describe the major steps for designing the JavaScript function(s) and how you tested them. Add some screenshots of the output

first, I need to fetch data from the link, and I need to check whether the access is available and the data is valid, if not, it will throw exceptions.

Next step is render the table using this data.

As for how I test them, I can use console, for example, I can print the error when catch them. And I can print data to check it structure and whether it’s valid.





## Task 6

Describe the major steps for designing the JavaScript function(s) and how you tested them. Add some screenshots of the output. Describe the difference between “.then” and “await” in terms of program flow

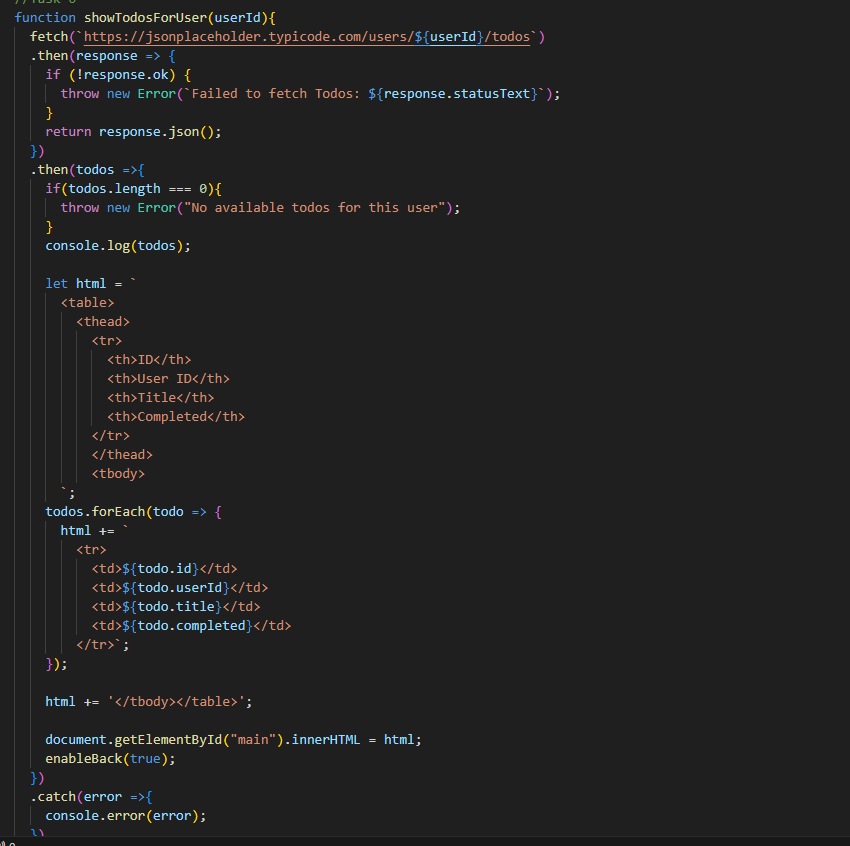
Difference: “await” pauses async functions until Promises resolve, simplifying code, while “then” chains operations on Promises, allowing sequential execution.

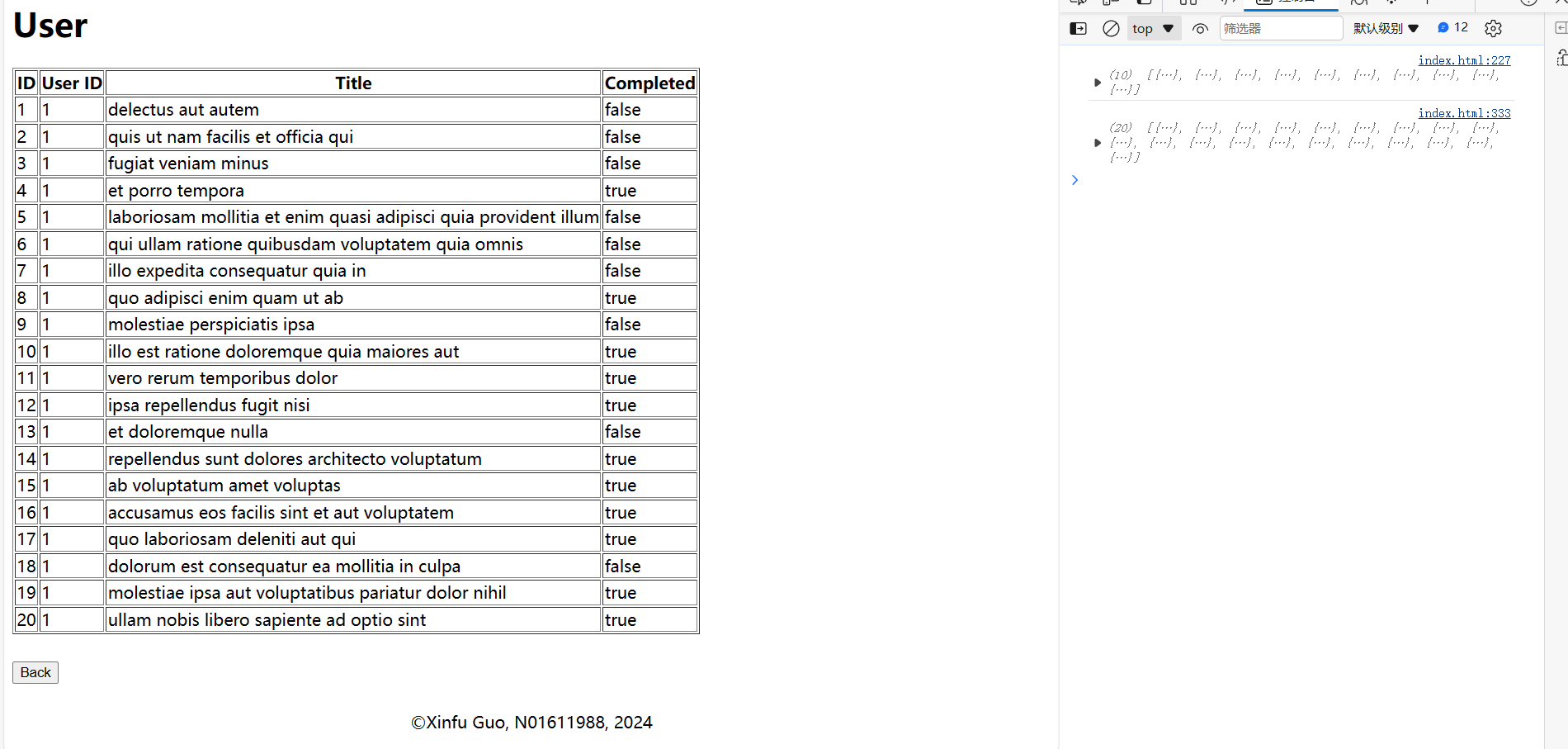
Use fetch to get data, and its syntax is different from “await”;

Use “then” and in next block to check data.

Use “then” and in next block to render table using todos data.

Use “catch” to deal with error.

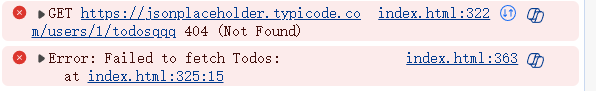


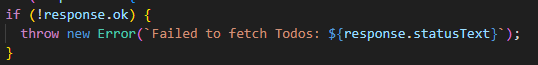


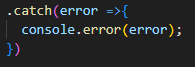
## Task 7

Describe how you simulated any errors. Include screenshots of an error being triggered and reported

I use try catch block or “.catch”, when there is error, I use console to print this error.



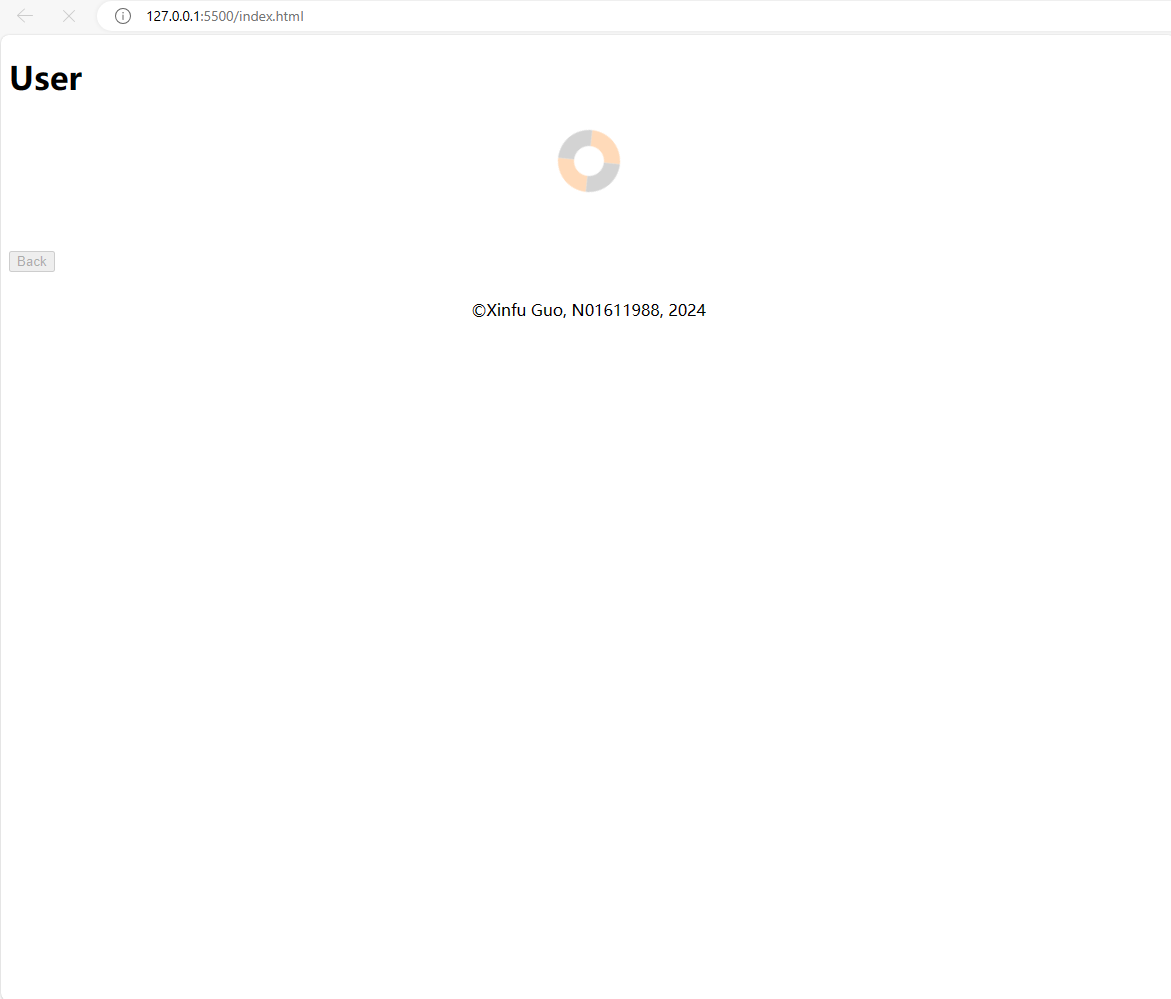




## Task 8

Describe why it is a good idea to show a loading indicator. Show a screenshot of the loading indicator

Displaying a loading indicator improves the user experience because it indicates to the user that the program is being processed, preventing the user from thinking that the application has stopped responding. It provides feedback to the user if there is a problem with the network connection and data is slow to be fetched by http or fetch, for example.



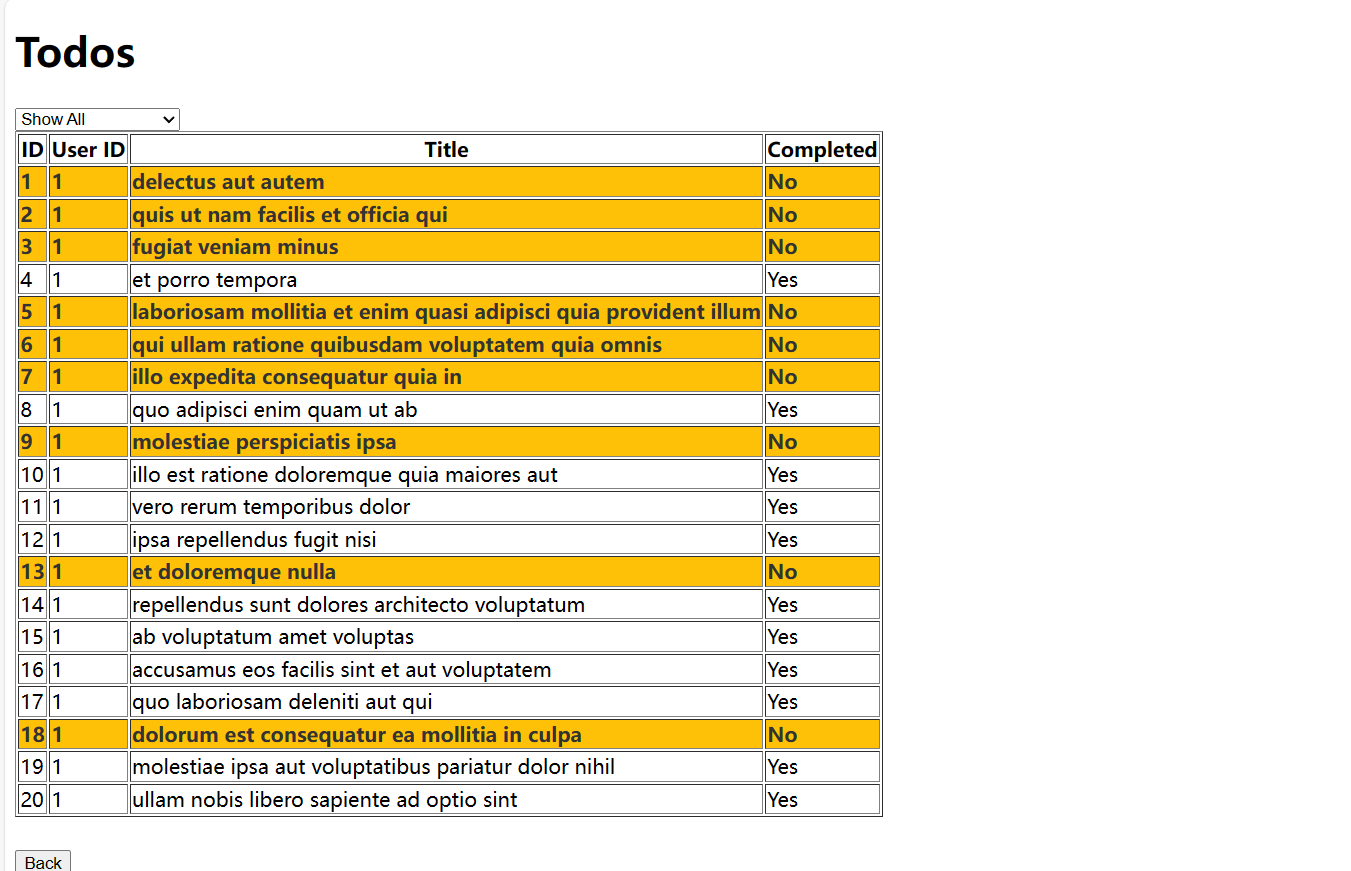
## Task 9

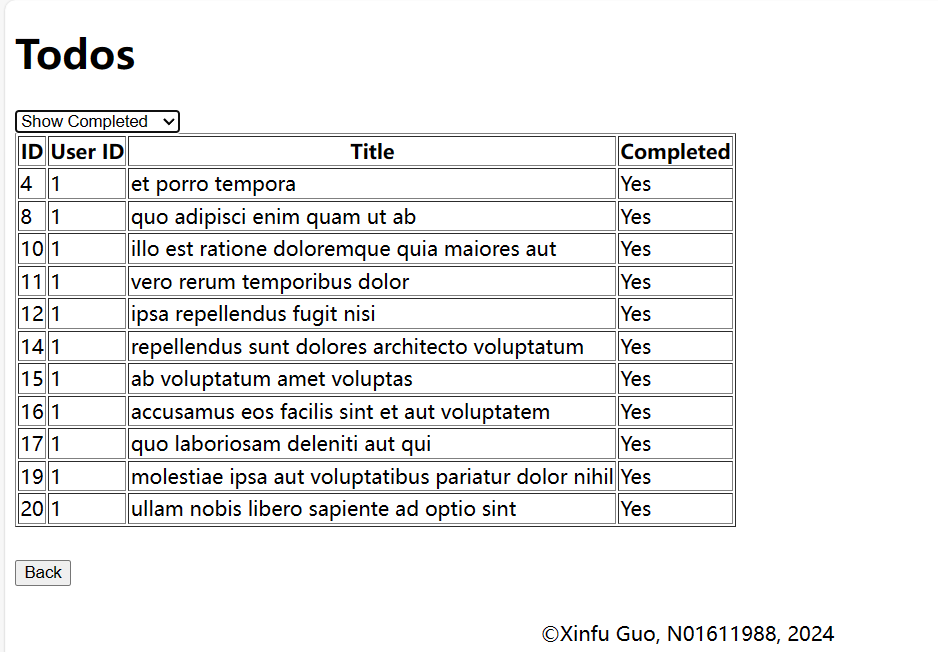
Show a screenshot of the highlight

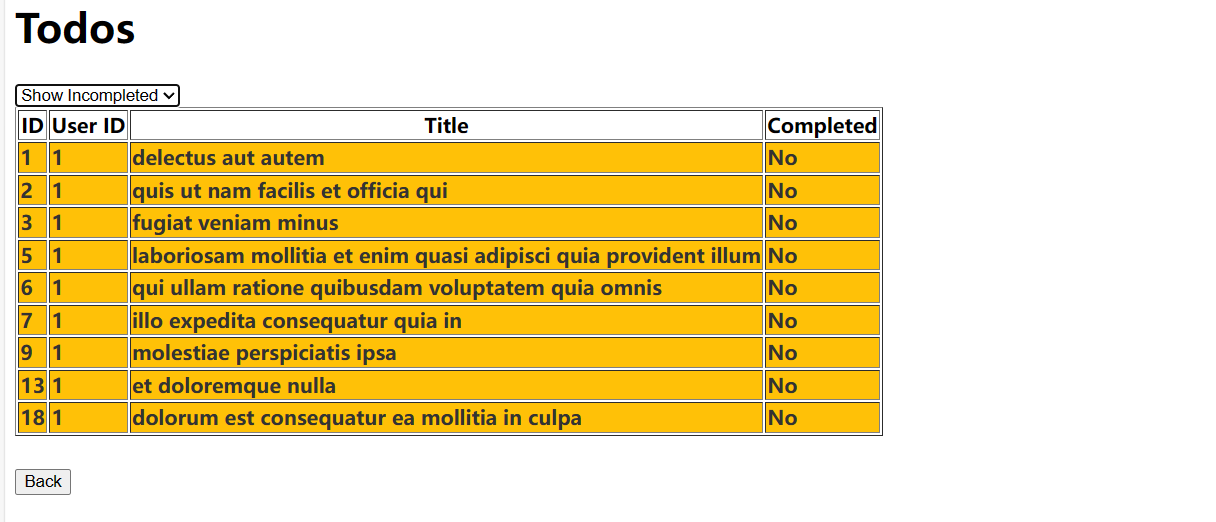
 

## Task 10

Show a screenshot of each option being tested

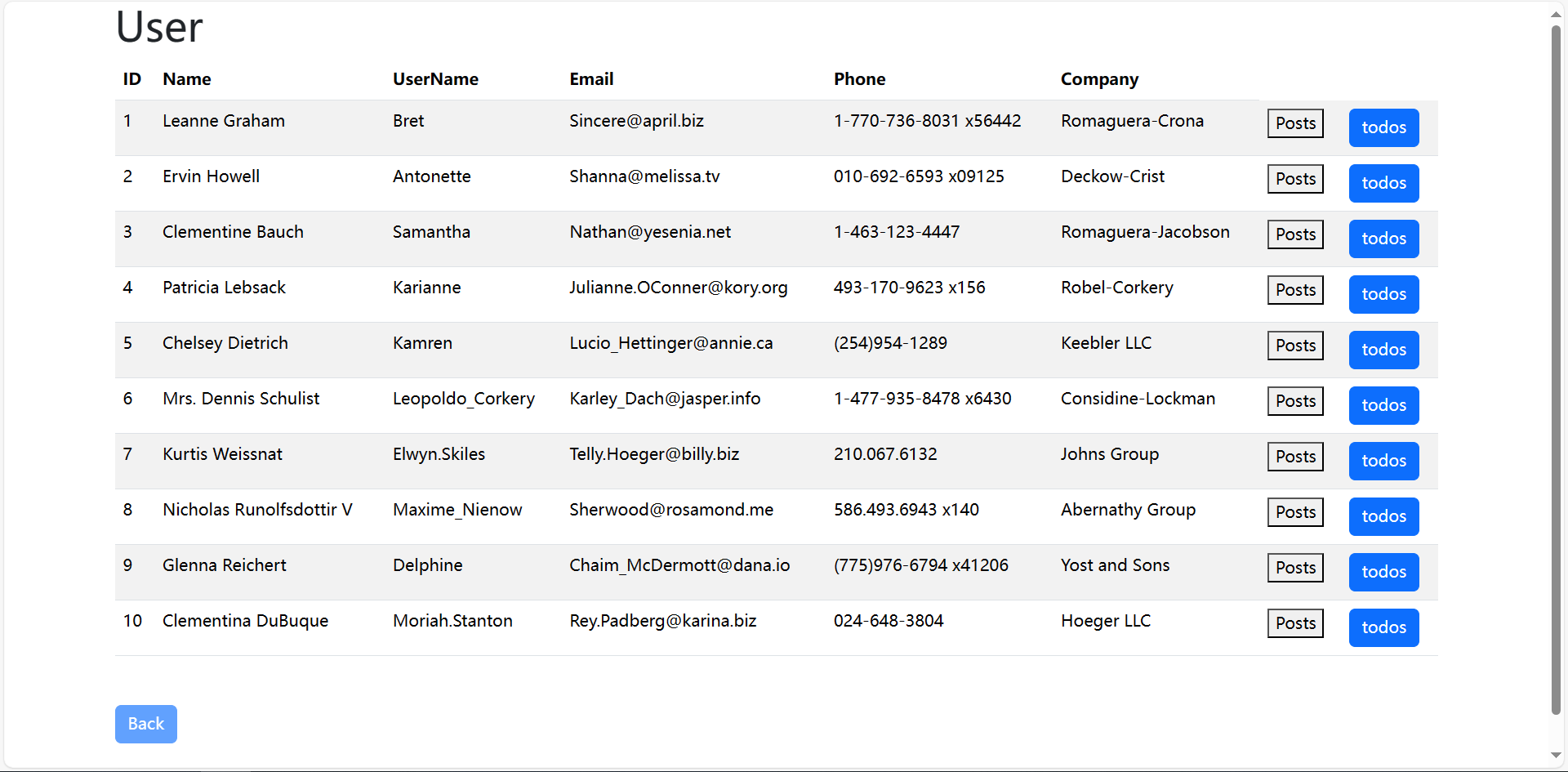


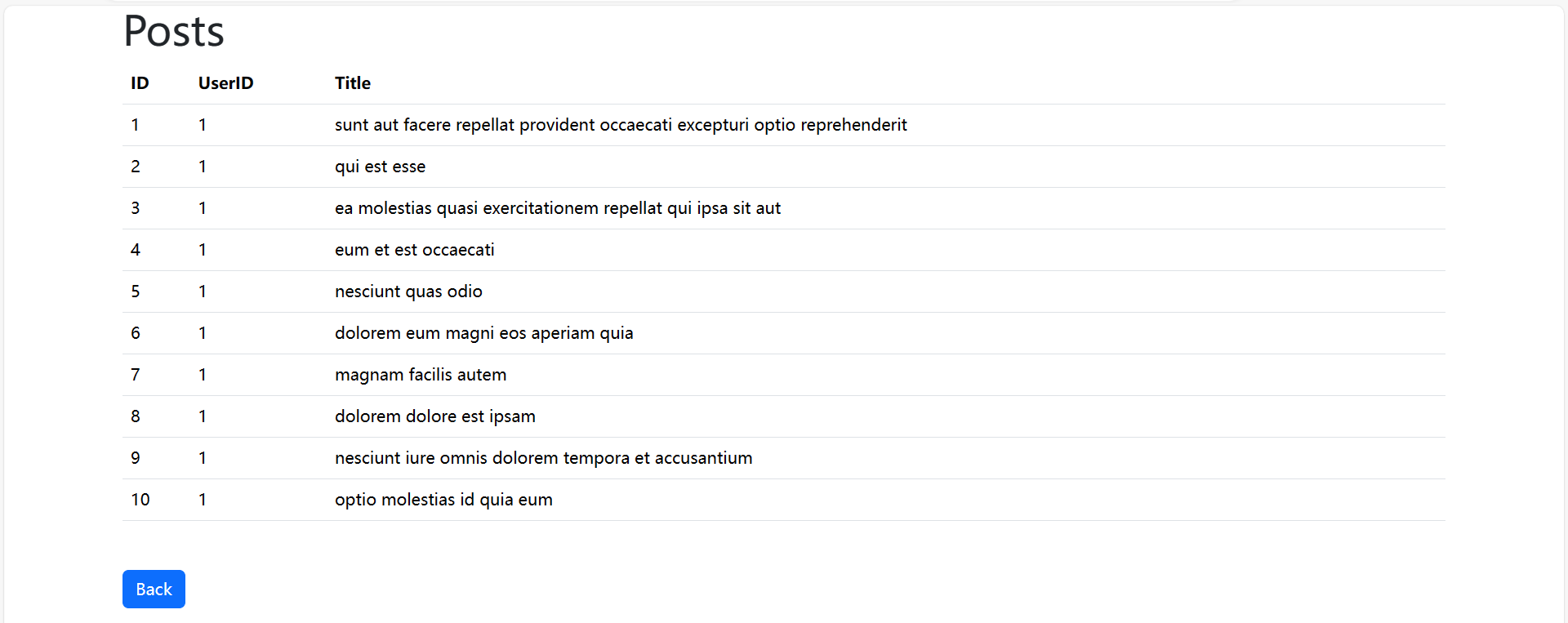


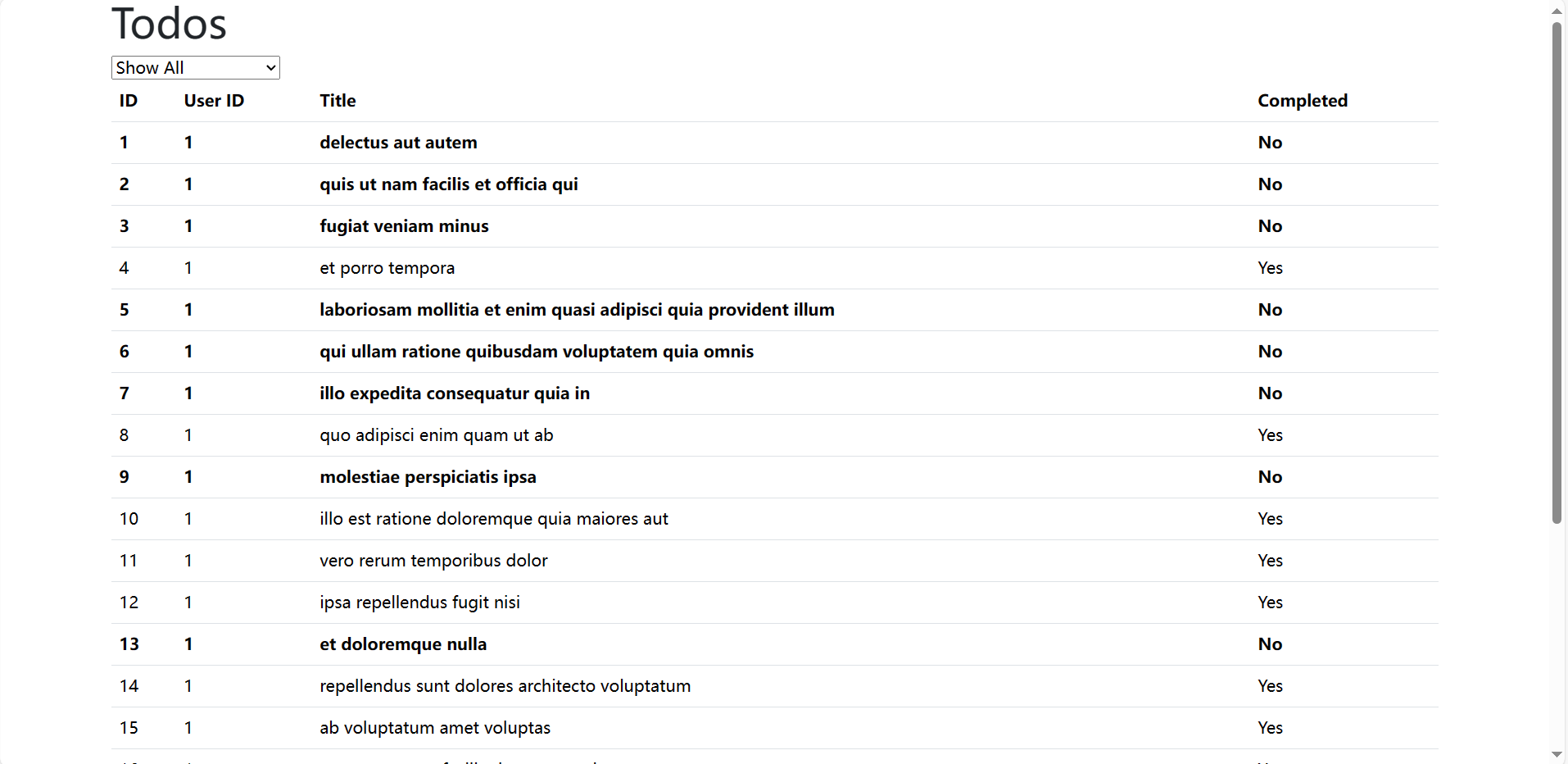


## Task 11

Show a screenshot of your UI updates







## Summary

Summarize your experience while working on this project. Include notes on any challenges you faced and if/how you resolved them. Did anything take longer than you thought? Was anything easier than you thought?

I think it’s a really good chance to practice and I can review JQuery, Fetch, Promise and error handler at the same time. It’s not easy because it covers lots of knowledges we learnt in this term. It’s makes me well-prepared for my final exam.

It took longer than I thought especially JQuery, I need to reach the slide and do some research which I don’t learn well. For example, in Task 10, I didn’t know how to achieve it.

All in all, it’s nice project, which make me learn a lot.