Promise Lab blank (1)

Instructions:

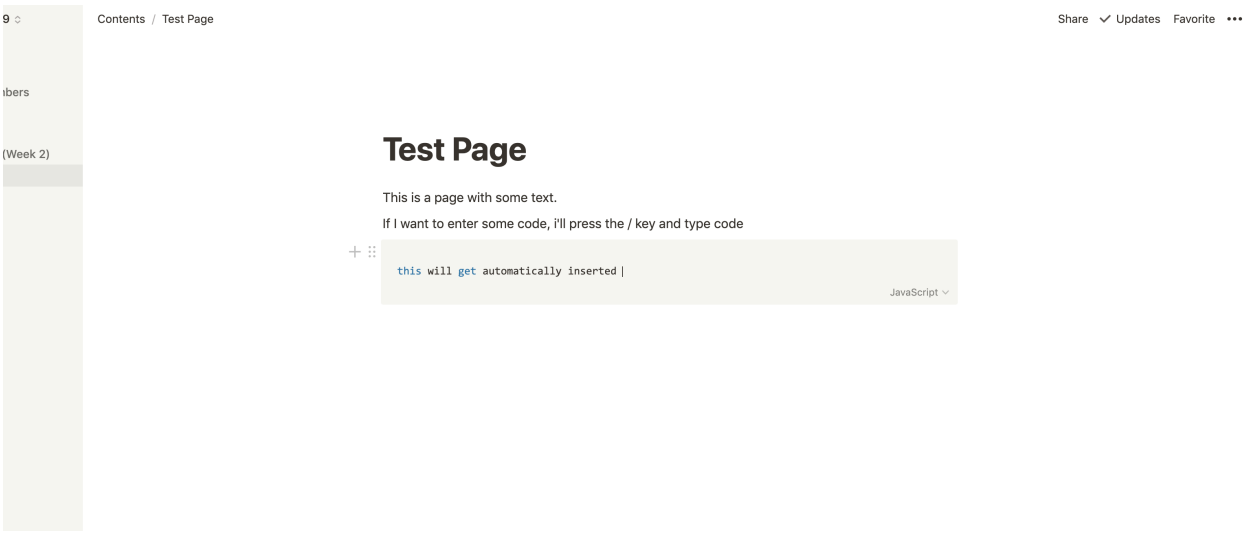
The following lab is due **10 minutes *before* the start of your next class with me. Please use your given name + preferred name if they are different:**

Submission Process:

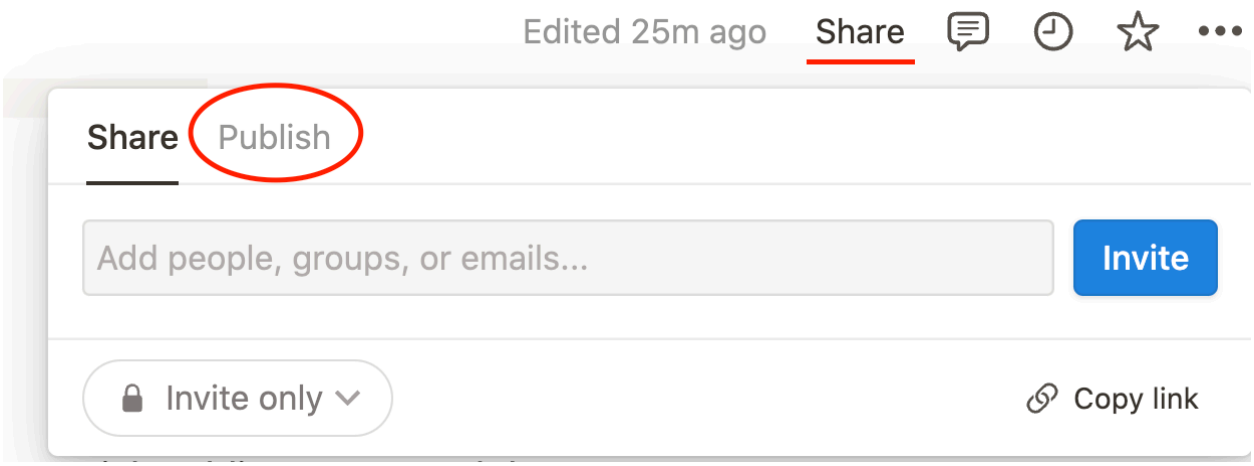
Notion is a tool which allows you to easily paste code and submit it to me. Go to https://www.notion.so/ to make an account, and then once logged into notion, create a page which you will use to answer the questions in this challenge.

You will see an area in the top right corner that says "Duplicate". Click this.

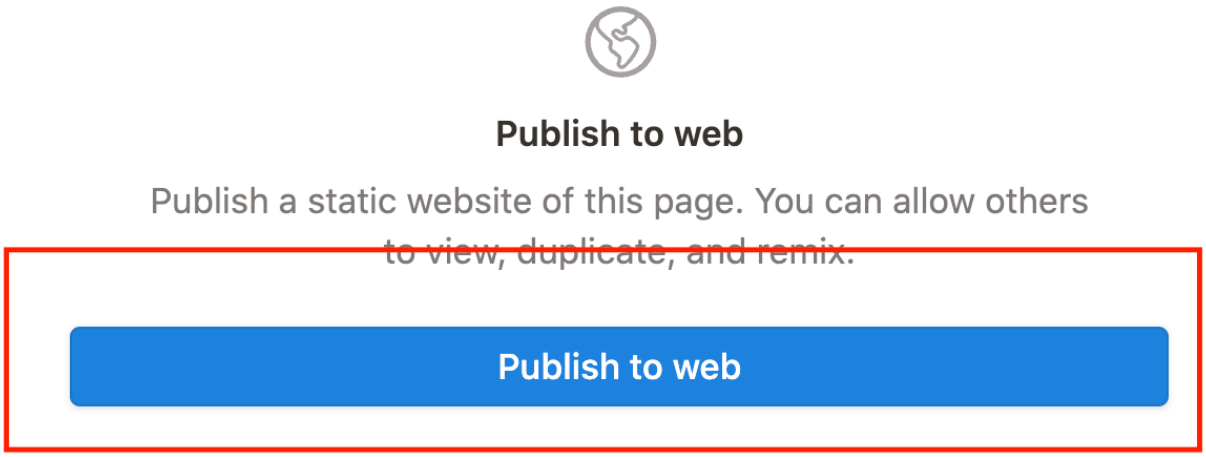
You will now have a copy of the Lab instructions within your Notion account. Fill in the answers to the questions.



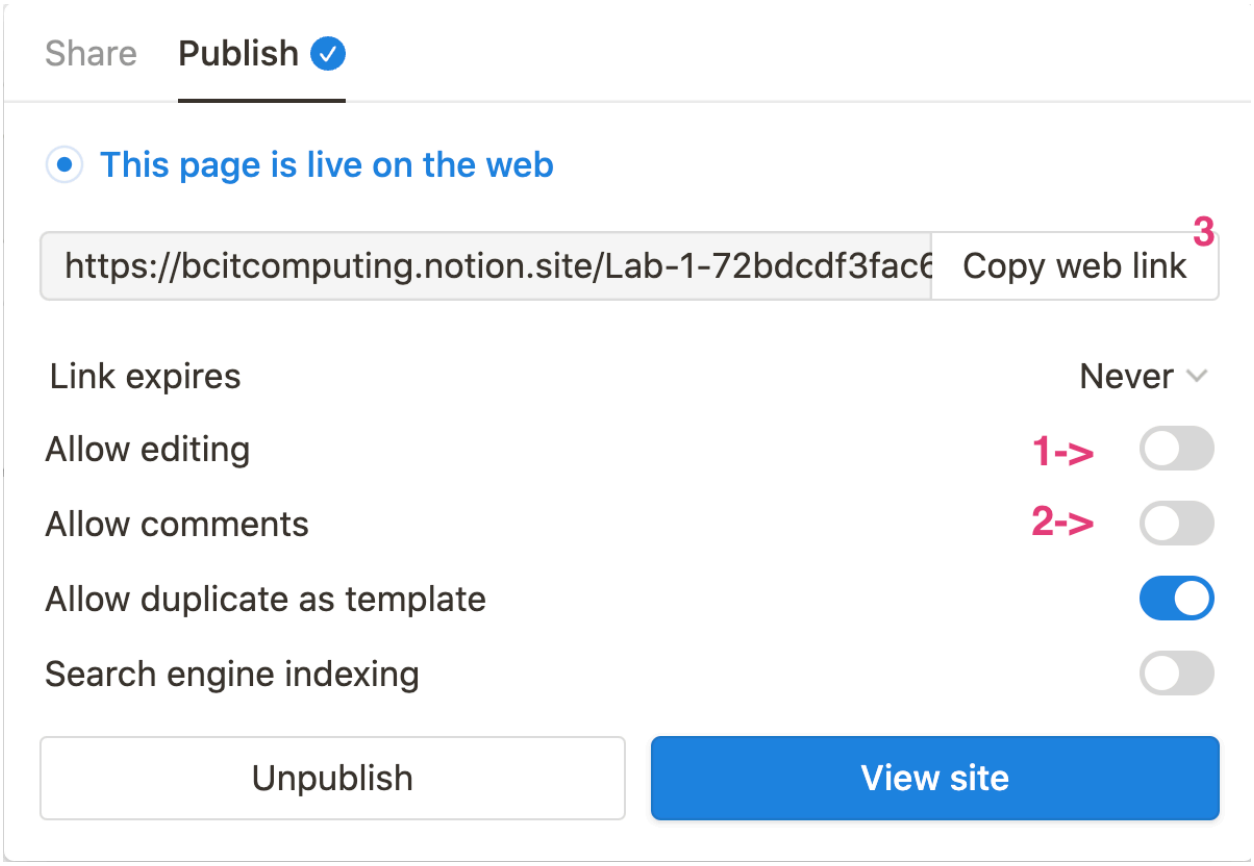
After you are ready, click the "share" button in the top right corner. Change tabs to the Publish Tab.



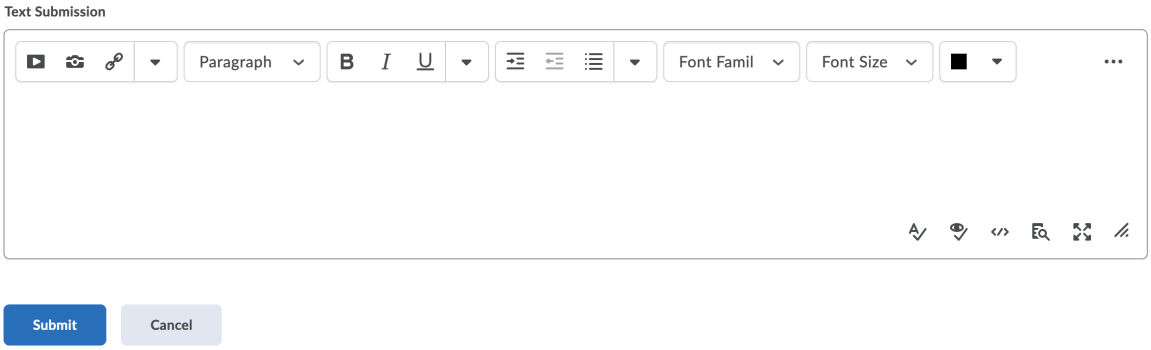
Click the **Publish to web** button.



Next, turn on **Allow editing & Allow comments. Then click Copy web link**:



Lastly, go to the **Activities > Assignments** section of learning hub, click the assignment, and then paste the notion link you copied into the text box. Click Submit.



Failing to follow these instructions will result in no credit being awarded for the lab.

> ⚠️ You should use **promises** for this question, not standard Node.js callbacks.

Your task is to convert a CSV containing items on a restauraunt menu to a TXT file. A CSV file is a **comma seperated values** file, where each line represents one entry, and each part of the entry is seperated by a comma.

To get started, create a file (with your mouse, manually) in VS Code called `menu.csv` and paste the following contents into your VS Code window:

```
lunch,bento box b - sashimi,box combo,$9.59
dinner,vegetable sushi,6 rolls,$3.50
dinner,tuna roll,3 rolls,$4.50
```

```
dinner,roe,2 rolls,$3.95
lunch,bento box a - chicken teriyaki,box combo,$8.59
```

Your Node.js program needs to read the file above, and write it to a file called `menu.txt`. When you write it to a text file, it should look close to what you see below.

(Note: The price of each time is slightly more expensive than what you see in the CSV file. Take this price increase into account by multiplying each item's price by 1.8 before writing it to menu.txt. Make sure prices are formatted to 2 decimal places. (Ex - Don't have $15.4637)

```
* Lunch Items *
$15.46  bento box a - chicken teriyaki, box combo
$17.26 bento box b – sashimi, box combo

* Dinner Items *
 $7.11   roe, 2 rolls
 $8.10  tuna roll, 3 rolls
 $6.30 vegetable sushi, 6 rolls
```

Heads up: The CSV ***could change*** to look like the following (note that we added dessert at the end):

```
lunch,bento box b - sashimi,box combo,$9.59
dinner,vegetable sushi,6 rolls,$3.50
dinner,tuna roll,3 rolls,$4.50
dinner,roe,2 rolls,$3.95
lunch,bento box a - chicken teriyaki,box combo,$8.59
dessert,cheesecake,1 slice,$8.00
```

which would require your program to output a text file like so:

```
* Lunch Items *
$15.46 bento box a - chicken teriyaki, box combo
$17.26 bento box b – sashimi, box combo

* Dinner Items *
 $7.11   roe, 2 rolls
 $8.10   tuna roll, 3 rolls
 $6.30  vegetable sushi, 6 rolls

* Dessert Items *
 $14.40    cheesecake, 1 slice
```

In other words, you cannot assume that the CSV will always contain either Lunch or Dinner. The category could be anything (ex, Dessert, Drinks, Appetizers, etc). Here is what you ***can*** guarantee:

The following structure will always hold true:

```
[Meal Type] [Meal Name] [Meal Quantity] [Price]
```

So looking at:

```
lunch,bento box b - sashimi,box combo,$9.59
dinner,vegetable sushi,6 rolls,$3.50
dinner,tuna roll,3 rolls,$4.50
dinner,roe,2 rolls,$3.95
lunch,bento box a - chicken teriyaki,box combo,$8.59
dessert,cheesecake,1 slice,$8.00
```

You'll notice that the first word in each entry is always the meal type (lunch, dinner, dessert), and the second word is always the meal name (bento box b - sashimi, vegetable sushi, tuna roll, roe, bento box a - chicken teriyaki, cheesecake), and the third word is always the quantity, and the last part is the price.

Your Solution:

(PASTE YOUR CODE TO YOUR SOLUTION HERE)

📌 Please include in your solution some code that calls your functions and shows them running in the correct sequence (ex - reads the csv file, then does parsing and writes to the txt file, and then shows a message saying “Program Complete”. Remember to **break your code up into smaller functions and not have one massive function that does everything!**