# Lilly Technical Challenge Documentation Template

## Approach and Objectives

* I started with the first objective, displaying the sample data on the screen by fetching the data from the ‘get medicines’ API that was already given, as all the other objectives fed off that one. I researched online1 how to fetch data and pass the data into a function to display the data in a table, and I found it straightforward to understand.
* I completed the second objective at the same time as it was essential to the functionality of the first. I added a fallback mechanism for the name and the price to recognise when the field was empty/null.
* I then attempted objective four which involved bringing the table I’d created in the JavaScript file to life in the html and CSS files.
* Objective three was the toughest because I wasn’t familiar with form submission in JavaScript. Using various online tutorials, I learnt how useful event.target is in its ability to pinpoint elements and that it could be used to attach to the form element 2, 3. . It was tough at first, but I was proud upon getting this objective working.

## Problems Faced

* I didn’t include error handling for network issues when I coded the fetch part of the JavaScript. I didn’t realise it from testing, but only when reading through websites to help did I spot it would be better coding practice to include.
* I tried running the html without the fallback mechanisms on the name and price. It did not cause a crash, it just left an empty table cell, like the below image. I learnt this is because the DOM handles undefined and null gracefully by treating them as empty strings.

A screenshot of a phone

Description automatically generated

## Evaluation

I spent more time than I was expected to on the challenge. I hadn’t done too much JavaScript in my studies so far, so I had to spend some extra time researching online, but I was proud of the functionality in the end. What I got through went well and was successful but took me some time to get right. I decided to learn how to dynamically add data to a table, instead of use bullet points, which I was proud I was able to achieve.

Given more time, I would:

* Attempt the optional objective of creating a backend function to average the prices of all the medicines.
* Improve the design. I do feel it is user friendly with the table, but I still feel improvements could be made, such as adding padding around the table to reduce the amount of white space.

I learnt a lot from the challenge and I’m looking forward to being able to discuss more about my code with you.

1 <https://www.geeksforgeeks.org/how-to-use-the-javascript-fetch-api-to-get-data/>

2 <https://developer.mozilla.org/en-US/docs/Web/API/HTMLFormElement/submit_event>

3 <https://javascript.info/forms-submit>