# Lilly Technical Challenge Documentation Template

*This documentation template serves as a place for you to discuss how you approached this challenge, any issues you faced & how you overcame them, or any other points that you feel would be relevant for the interviewers to know. The text in italics is here to guide you - feel free to remove it once you fill out each section!*

***Not every section in this document is required. This is just a template to help get you started. Feel free to add or remove sections as you feel necessary.***

## Approach

*How did you approach this challenge? Did you work through the objectives in any particular order? If so, why? Did you utilize any external resources, such as tutorials, guides, or other materials?*

My approach to this challenge was to systematically break down the challenge into its objectives one by one, and further breaking down each objective into smaller tasks to make each step more manageable.

Luckily in my last semester of university I had experience with JavaScript and API requests for making a responsive website that displays data from JSON formatting so I went about creating my JS functions for displaying, updating, adding and deleting the json.data and integrating it into my front end html files.

I made sure to use specific and concise class and id naming techniques so that there was no clutter when it came to integrating the front end and back end logic seamlessly.

Once I had the required javascript functions I could create a skeleton for my html files, to make it look nice and user friendly. I referred to my project files on my github repositories for my previous project on how to do some of this functionality as a lot of what I had done was a bit vague in my memory so I had to refer back to some of the syntax and logic implementation.

*I rearranged the DIV elements to do with adding a medicine, calculating the average and viewing the medicines because I initially put the list of medicines first and I realised when viewing the website under different viewports that to get to adding a medication or calculating the average price the user would have to scroll down all the medicines first. With only a few medicines in the JSON.data it’s only a small amount of scrolling, but I realised if that were scaled up to a realistic database of medicines that’s potentially thousands of data entries needed to scroll through before the user can interact with the websites functionalities.*

## Objectives - Innovative Solutions

*For the challenge objectives, did you do anything in a particular way that you want to discuss? Is there anything you're particularly proud of that you want to highlight? Did you attempt some objectives multiple times, or go back and re-write particular sections of code? If so, why? Use this space to document any key points you'd like to tell us about.*

The one thing in particular that I was proud of, was my input validation for the price accepted from the user: both when they add a medicine or when they update the price. Originally, it was accepting and saving prices beyond two decimal points. So I wanted a way to only allow for the user to input two decimal points. Initially, I wanted to do the input validation FORM side, but I realised if I could limit the input field for the price on the user side, I could address the problem as early as possible and stop any invalid data from reaching the back end at all.   
  
A screen shot of a computer code

Description automatically generated

I did this by creating an event listener for user input at the medicine price element id, and checked if the inputted value contained a point. If so, split the input value into integer and decimal parts and then if the decimal part is greater than 2 it is reduced down only to two decimal points.

## Problems Faced

*Use this space to document and discuss any issues you faced while undertaking this challenge and how you solved them. We recommend doing this proactively as you experience and resolve the issues - make sure you don't forget! (Screenshots are helpful, though not required).*

As mentioned in innovate solutions, the user input validation for price decimal points.

I got help with this problem from this website where someone had a similar problem

[Restrict user to enter only two decimal poins | Support Center](https://support.pega.com/question/restrict-user-enter-only-two-decimal-poins)

Trying to allow the user to edit the price inline without linking to a separate html file designed for editing the medicine price, proved a particular challenge. To do this I introduced a togglable editable field,

A computer screen shot of a program code

Description automatically generated

BY checking if the priceElement is editable if it isn’t make it editable then it takes the new values inputted by the user and when it the save icon is clicked it calles the savePrice function which calls the update function and updates the medicine using the name from the medicine.name and new price.

Another problem I faced was message displays. All messages displayed when the user interacts with the website is server side. I wasn’t able to, in the time given, figure out how to display user friendly looking messages.

## Evaluation

*How did you feel about the challenge overall? Did some parts go better than others? Did you run out of time? If you were to do this again, and were given more time, what would you do differently?*

I felt the challenge was perfect for the scope of the interview. As it was explicitly stated in the brief that it was not expected of us to go beyond an hour in implementing our solutions, it meant that the challenge scope did not feel daunting when as a second year university student I have a lot of pressure to be completing my course-works.

I feel it was hard enough to provide a challenge and provide insight into the problem solving techniques used, but not hard enough that it felt unapproachable to someone who was unfamiliar with the technologies used in the challenge.  
  
I did run out of time, but I still finished the challenge and that was because I was happy to spend longer fine tunning and debugging any problems I was facing and trying to present a better project. I feel there was suitable time to just barely implement the functionality required if strictly working within the time frame.

What I would do differently, if given another opportunity to do this challenge, I would implement additional checks in the back end just to ensure watertight data integrity.

I would create more user-friendly display messages for things like updating price or adding medication as right now its janky and it’s the server logic that is displaying the messages, which is an ugly design.

I would check for duplicate data entry and have input validation for those instances.

I would add features such as Search and Sort features and Authentication for editing and deleting.

I would use a database instead of data.json for more scalability and efficiency and better data consistency through transactions.

I would improve the aesthetics by using a framework like bootstrap.