# Lilly Technical Challenge Documentation Template

## Approach

***How did you approach this challenge?***

The first step that I took in tackling the challenge was to go through the files to familiarize myself with structure of the repository, the existing implementation and the available dataset.

During this analysis, I found that there were several issues such as missing error handling in the backend configuration, improper null value handling and no validations on the data. I also realized that there was no frontend implementation as such, apart from a basic template.

I decided to approach the challenge by fixing the API issues on the backend before moving to the frontend implementation.

I added the API to find average price of all medicines as it is a simple addition to the existing get methods.

***Did you work through the objectives in any particular order? If so, why?***

Yes, I started to work through the backend code, concentrating on fixing the error handling and optimization issues, as this was the main goal of the exercise. Without these fixes, the frontend would easily break because of the data inconsistency.

Next, I worked on the basic frontend functionality to display the medicine data in a more customer-friendly manner, Allowing the end user to check the medicine information.

I then added the frontend functionality to update, delete and create new medicines on the webpage.

Finally, I implemented the backend functionality to fetch average price of all medicines and display it on the webpage.

I then worked on optimizing the code and adding further validations to prevent any edge cases from breaking the code.

***Did you utilize any external resources, such as tutorials, guides, or other materials?***

Yes, I made use of some online resources such as Stack Overflow, Python docs, jquery Docs and Google search engine for some help with HTTP Exceptions and the relevant status codes, CORS errors and Git issues.

## 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.***

I decided to display the medicine data to the end user as a table, allowing them to view the prices and also to delete and update the medicines.

I handled the errors and exceptions gracefully without the frontend or server breaking for any of the inputs. I ran tests from both user inputs on the frontend as well as direct inputs using postman to ensure the server handled all the negative scenarios.

I iteratively developed each frontend functionality, ensuring that they worked properly before moving on to the next feature.

I was also able to develop the average price functionality and display the average price of all the medicines on the frontend. It is dynamic and varies with any changes in the medicine prices

## 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).***

One of the problems that I ran into while developing the frontend was CORS issue. I had used the $.post jQuery method which was getting blocked by CORS because it did not have the appropriate content type header. I searched the net for solutions and found that using $.ajax method actually allows us to configure the request headers and thus allowed me to set the required content type.

## 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?***

The overall challenge was interesting and adequately allowed me to showcase my development skills. I mostly concentrated on improving the backend functionality and developing a basic frontend functionality rather than spending too much time on styling. I used bootstrap classes to save effort. I also developed the frontend as a simple table displaying the JSON data.  
If there is enough time and resources, there are certainly many aspects of the code that can be improved.

The way we are storing data into a Json file and the use of .seek(0) can cause concurrent writing issues and potential loss of data. We must ideally use file locking or a database to store data. This also allows us to secure the data with user validations.

I used a simple page refresh function to display changes to data in a dynamic sense. But ideally, we can use React.js for such use cases.

I thought of implementing a sort functionality that allows the user to sort the table according to price or medicine name, with a pagination to restrict table content to 10-15 medicines per page, but did not implement these due to time constraints.