# Lilly Technical Challenge Documentation

## Approach

For this challenge, I started by reviewing and understanding the challenge objective. I prioritised setting up the project environment first, ensuring python was installed etc. My approach was in the following order: implement, test and refine any features one at a time to ensure smooth integration and functionality. I planned to implement the basic CRUD (Create, Read, Update, Delete) operations first as they were the main functionalities. I utilised features provided in JavaScript, CSS and HTML to make the frontend interface more user friendly. I referred to external resources such as w3schools for CSS, HTML and JavaScript, Font Awesome for the icons and Mozilla Foundation for JavaScript.

(1)<https://www.w3schools.com/css/> (2)<https://www.w3schools.com/html/> (3)<https://www.w3schools.com/js/> (4)<https://docs.fontawesome.com/web/use-with/python-django> (5)<https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction> (6)<https://developer.mozilla.org/en-US/docs/Web/API/EventTarget/addEventListener>

## Objectives - Innovative Solutions

I implemented features with a focus on user experience and branding. One innovative solution I implemented was the Lilly logo, which added a personalised touch to the webpage. I also spent time experimenting with the icons for buttons and headers, looking at different variations available until I found a set that gave clear indication for the user. I also introduced a progress bar to enhance the user’s understanding of ongoing processes, this allows for a smoother and more interactive experience. I had to rework some sections, especially the confirmation pop-up window for deletion. Initially, it didn’t function as expected, so I optimised the timing to allow the user to interact. While it didn’t fully resolve the issue, the adjustments improved and brought me close to the desired outcome.

## Problems Faced

During the challenge, I faced several issues:

1. **Double $ Signs in Medicine List:** Initially, the medicine list displayed prices with double dollar signs. I correct this by *changing listItem.textContent = ${name} - $${price};* to *listItem.textContent = ${name} - ${price};*
2. **Price as Integers Only:** The price field only accepted integers. I updated the input field in index.html to include *step = “0.01”* and parsed prices as floats in script.js.
3. **Medicine List Not Displaying:** When trying to add the quarterly report, it caused the list to disappear. I fixed this by ensuring in the main.py and script.js file, the url endpoint ended in [*/quarterly-report*](http://localhost:8000/quarterly-report) was the same.
4. **Add, Update, Delete Functions Not Working:** I ensured that the form submissions correctly captured and sent data to the backend using *FormData*.
5. **Quarterly Average Not Displaying:** The report didn’t display initially because I hadn’t handled the data properly. I added *toFixed(2)* to ensure correct decimal formatting and verified that the backend provided valid JSON data.
6. **Lilly Picture Not Showing**: The image didn’t display because I hadn’t saved the image file in the same directory as the HTML file. Moving the image file fixed this.
7. **Delete Confirmation Popup Issue**: I implemented a modal for delete confirmation, but it only showed for a few seconds and didn’t allow interaction. I realised that the form submission was causing the page to refresh immediately, so I added *event.preventDefault().* This allowed the modal to show up for a bit longer but still didn’t allow any interaction.

RAW – FORM DATA

I assumed that the code was using form-data for the POST /create endpoint because of the way the parameters were defined in the FastAPI endpoint. In this case, name and price are being taken from Form(...), which specifically indicates that the parameters are expected to come as form-data rather than raw JSON.

*A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.*

## A screenshot of a medical form AI-generated content may be incorrect.A screenshot of a medical form AI-generated content may be incorrect.

## A screenshot of a medical form AI-generated content may be incorrect.

## Evaluation

Overall, I found the challenge both stimulating and rewarding. While some parts, like the modal functionality, required more effect and multiple attempts, other aspects, such as improving the user design, went smoothly. I did run into time constraints while trying to refine the UI and additional features, but I am satisfied with the core functionality implemented.

If given more time, I would enhance the UI with more interactive elements, implement comprehensive error handling and refine the additional features.