

# Approach

---

For the challenge, I started with analyzing the objectives and identified the key requirements like fetching data, handling invalid data and implementing user friendly UI functionalities.

I approached the objectives in the following order:

1. **Fetchning and Displaying Medicines:** Wrote Javascript to ensure data is retrieved from backend and displayed properly.
2. **Handling Invalid Data:** Made changes to the display of medicines and removed or displayed appropriate message for the invalid data.
3. **Adding functionality to update/delete medicines:** Made functions to add, update and delete medicines with user friendly interface while also handling the invalid inputs.
4. **UI Enhancements:** Displayed each medicine data with cards.

I was not familiar with the FastAPI before this challenge, So I learned its basics especially how to handle GET, POST and DELETE requests. Additionally, while I had experience with fetching data using GET, I had to learn how to properly use POST and DELETE requests using JavaScript.

To learn these referred to:

[geeksforgeeks FastAPI Documentation](#)

[freeCodeCamp JavaScript Fetch API For Beginners](#)

## Objectives - Innovative Solutions

---

### Handling Data Errors and efficient data handling:

- Implemented logic to check for missing medicine names and prices.
- Created a dictionary to store medicines for quick lookups and make it efficient by not sending a request every time data is required.

### User-Friendly UI:

- Used Card to display medicine with update and delete button so that user don't need to manually add medicine names to delete medicine or update price of the medicines manually.
- Added search functionality so that user can easily look for the medicines they are looking for.

I revisited my approach a couple of times for the handling of API and UI. As I mentioned in my approach that I was not familiar with the FastAPI and the POST and Delete, I revisited that part multiple time when I learned new ways to handle these requests. For the UI I first displayed it in the tabular form but it was not very user friendly and I recently wrote code to display car in card form for my group project in university so I used that code with some modifications to display medicines in the card form.

# Problems Faced

---

## Learning to Use POST and DELETE in Fetch API:

**Issue:** I was only familiar with the GET request and never used the POST or DELETE with the Fetch API and while learning it from freeCodeCamp, I was not able to understand how to correctly implement correct headers and body format.

**Solution:** I learned it and applied from a YouTube video from NovelTech Media- Fetch API.

## FastAPI Returning 500 for Endpoint:

**Issue:** When making a fetch request to the "/averagePrice" the API returned a "500 Internal Server Error"

**Solution:**

- Found the problem for the error 500 that when it was processing data to calculate the average due to the invalid entries in the data.json file the processing failed every time returning error.
- Implemented data handling to skip the invalid entries.

# Evaluation

---

The challenge was great learning experience. With fun tasks like handling missing values and adding calculate average features requiring problem solving.

## What went well?

- Backend Integration – Successfully connected FastAPI with JavaScript.
- Error Handling – Ensured invalid data didn't crash the application.
- User-Friendly Features – Added search, add new, delete and update operations with clear messages and data handling.

## What could Be Improved if given more time?

- I have used location.reload() to reload the page to apply changes if given more time I would use state management to have a smooth experience.
- As there are not much medicines in the database. I am loading and displaying all medicines at once in cards form but I would like to change it and display some medicines on loading the page and add a load more button to load more medicines if needed. These will be much efficient when the dataset is larger.
- I would also add better UI styling and look at my usability engineering module concepts to increase the user experience.