Problem Statement

First, I read the software requirement specification as well. After watching it later, I thought, What did you want from me? Well, I found I would create a food delivery layout. Then I think I have a food delivery layout that I created in my early days when I started learning HTML, CSS, and JavaScript. That's it. The layout is final, and I think it's the best opportunity to transform a static layout into a dynamic layout. then start thinking about an API. How can I design this API? Because I see that authentication and authorization are very important parts of this type of application, I built this API with that in mind.

You asked me how I could approach this type of problem. It's simple: every API does a crud operation every time. So don't think too much and don't waste time; I just wrote my code from day 1. I first built API endpoints like signup and sign-in. create-food-item, update-food-item, delete-food-item on day 1, and head over in front to build static to dynamic. Before, I built it on plain CSS only, but this time I needed to integrate Tailwind CSS. So we need to recreate this whole layout in the Tailwind version. So I started to create this layout first in the static version with Tailwind, just because I'm new to Tailwind CSS. Then I integrated my backend API into the frontend. Now again, I'm stuck with another problem, which is state management in the React project. It's not a big problem; I can use the props drilling mechanism or use context-hook with the user-reducer, but I think it's the best time to introduce Redux and redux-toolkit because, after learning React, redux-toolkit is a common and essential tool for every developer for state management. After doing state management, I am facing another problem. That is payment integration. How can I integrate with this application? Then I go to Google and search for it, and they give me some options like Stripe, Razorpay, and Platforms. I spent two days learning how to integrate this into my application, and I built two prototypes for testing purposes and to see how payment integration works. I also completed the last requirement in SRS. In closing, I say it's the biggest self-learning journey in my life.