# Foreword:

At first, I need to thank you for checking out my code. I would really appreciate it if you ***could share your feedback with me regarding the coding challenge***, even if my interview process lead to my rejection. This coding challenge was a great learning opportunity for me and I don't know how to express my appreciation about It!

It is worthy to note that this coding challenge could be answered far better, but due to my illness during these two weeks, I could not concentrate enough of the task and could do my best job. However, I did according to what was mentioned in the assignment and prioritized quality over feature-completeness***. It was my first time with fastapi and the second serious project of my life time!*** Here is the link to the [***Github repository***](https://github.com/hmpsl99/online_shop).

I spent nearly 20 hours for the task and I dedicated the half of it to learning FastAPI in general. I used a book called *Building Python Web APIs with FastAPI: A Fast-paced Guide to Building High-performance, Robust Web APIs with Very Little Boilerplate Code* for getting the general idea and I watched various videos on Youtube and of course googling. If I spent more than half of this time doing these things I haven't lied!

# Description of the Problem:

I tried to solve the first problem mentioned in the assignment called *Shopping Cart.*

* " Create a Restful API that serves as stateless shopping cart backend. Handle all relevant requests such as adding a product to the shopping list , removing a product from the shopping list, etc. Authentication/Authorization would be necessary and we expect it to be handled based on JWT technology for all the services of the project"

# Description of the Solution:

Although I didn't have the time to implement the whole solution but I think I need to explain it all.

## Database and Database Models:

For simplicity reasons I use SQLite for my main and test database. Although in the similar project on github, redis was used but due to the fact that I don't have the working knowledge of using redis I couldn't use It. It could really come in handy while dealing with shopping baskets.

I had the experience of working with Django and Django ORM but dealing with SQLAlchemy was a new challenge for me. Because I had to do all the things manually while Django did them all by magic.

As you can see in the models.py in the code, we have 4 entities in this project:

* User
* Product
* BasketItem
* Basket

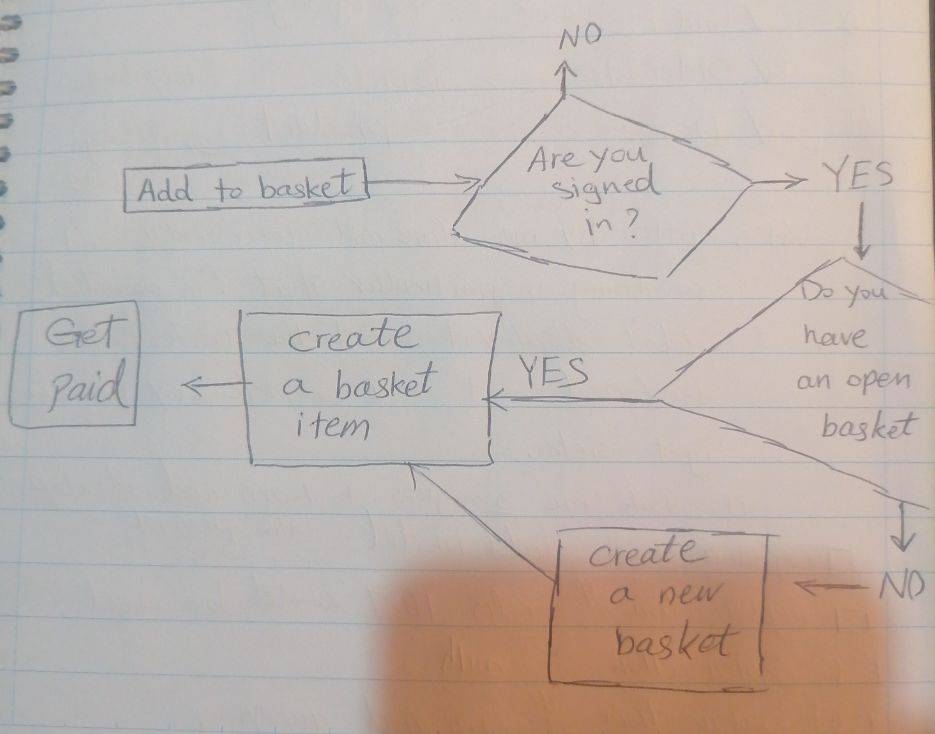
In which User had a one to many relationship with basket, and basket item had two one to many relationships with product and basket.

The reason I included basket Item was that in the scenario which a product changes, we don't want the previous information ie price to change. Each basket includes one or many basket items.

## Logic:

The Process that I had in mind was like this( which unfortunately didn't have time to implement):

Here is the general flow (off course I forgot to implement the Status model for the basket)



I implemented the routes related to product and user and authentication. The project only covered the backend and I think the architecture is MVC.

## Testing:

I implemented one unit test for one of the endpoints.