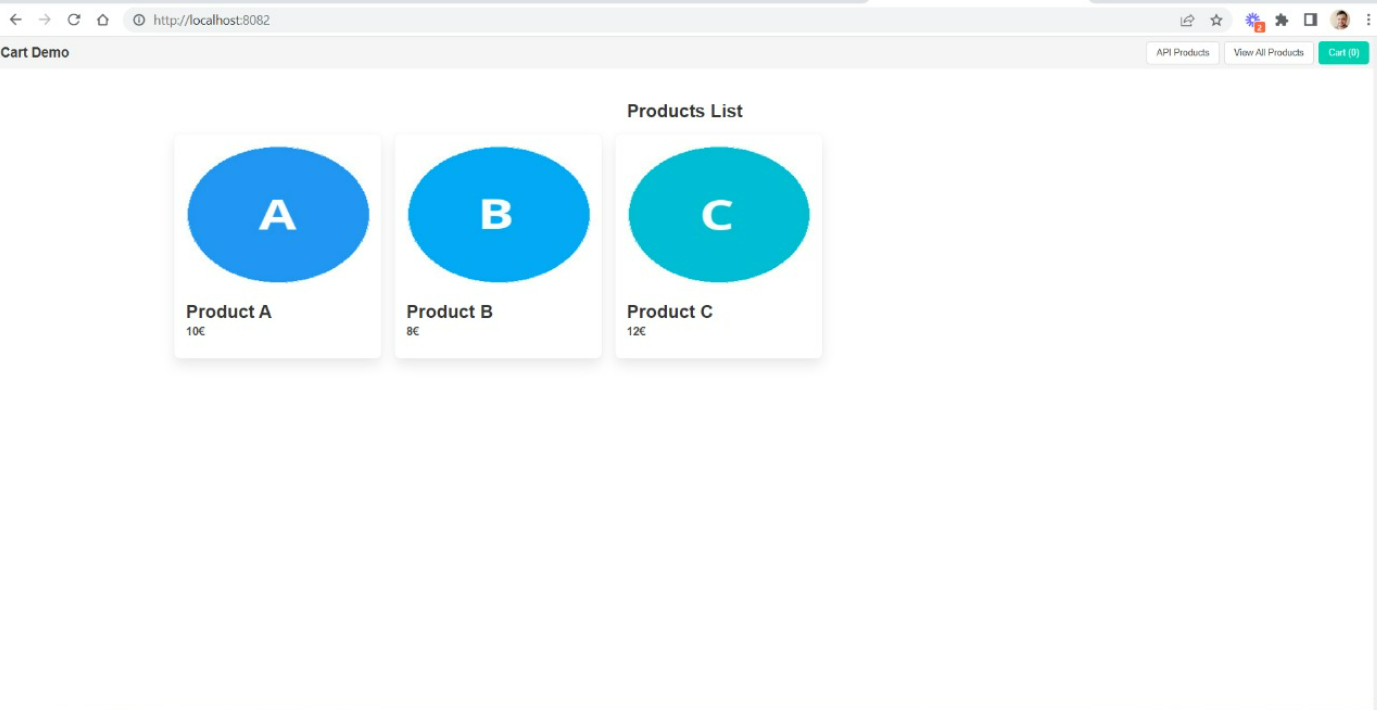
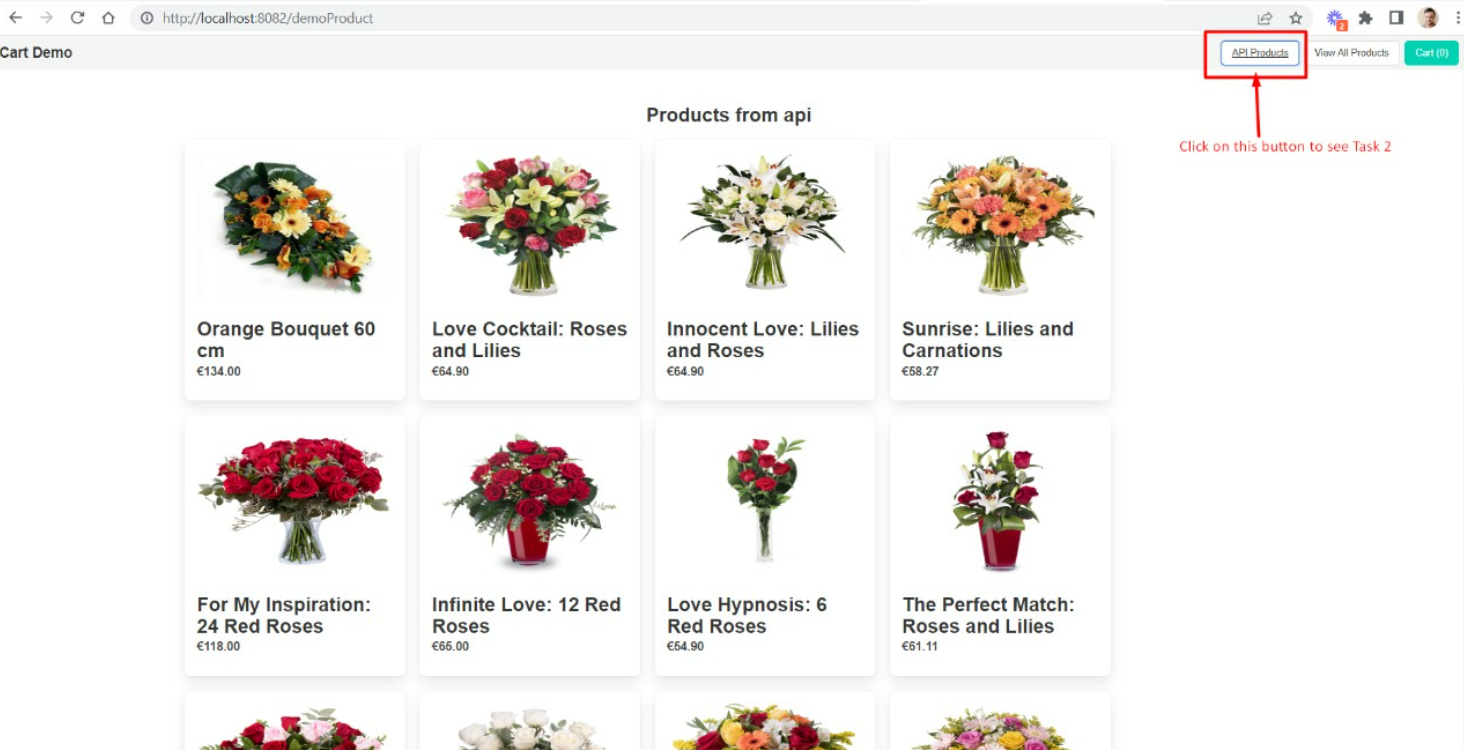
**Task 1:** Run the this [Github](https://github.com/pvnrg/shipping-cart-vue) project in your local by following steps given into readme.md file



**Task 2:** in the setup project you can see the API Product link in top right menu items, just click on it and you can see Grid View of product using given API in Task description. Also, by clicking on any grid item you can see the detail page of that item.



**Task 3**

**Question**: How would you build an asynchronous and distributed architecture, how would you ensure that no information was lost and in case there were failures, detect them?

**Answer:**

Async architect.

One system to receive [**rabbitmq**](https://www.rabbitmq.com/) events with payload.

System itself will store all events in db. So can be dispatched again if the first event failed.

Each service in distributed architecture will be synced by events.

[**Rabbitmq**](https://www.rabbitmq.com/) or similar messaging system can be used to dispatch and listen to events.

There should be one central system that listens to all events and stores them in its local db.

When any system reports event failure (by logs) similar event can be dispatched again from central monitoring system

This is a top level idea of how a system can work as a whole and sync data together with failure recovery.

I am sure that this can be improved when actual implementation starts.