Assignment: Building a system to manage the product list at Tiki.

Student name: Nguyen Quoc Hoang

Student ID: s3697305

Project name: Kiti – A Tiki’s ripoff ¯\\_(ツ)\_/¯  
Project repo: <https://github.com/hoangdesu/Kiti-fullstack-React-SpringBoot>

# Reflection

## What technology and architecture did you choose for?

For frontend, I use React.JS as the main framework for building the reactive user interface. The website data is provided by Spring Boot from the backend. The architecture used in this project is client-server where the Spring Boot provides API RESTful services for the frontend to fetch and display the data. For database, I use PostgreSQL to store and query data.

## Why did you select that technology and architecture?

React is one of the most popular frontend frameworks as of today thanks to its flexibility in building user web interface. On the other hand, Spring Boot is known as an enterprise level framework thanks for its robust power and high security. Meanwhile, PostgreSQL provides a strong and reliable database management system to store all my data into relational schemas.

## Why did you define such attributes for your products and organize such categories?

I have a relative long list of attributes, which has 10 key attributes I think which the most important for an ecommerce website. The id attribute is generated automatically by the server to avoid collision and also used to identify between products. The list includes:

* Id: long
* Product name: string
* Categories: string
* Price: double
* Discount: double
* Added date: LocalDate
* Rating: double
* Seller: string
* In stock: int

## Whatever arguments to convince your clients, your boss that your system is elegantly designed

Kiti is designed with a user experience-oriented approach in mind to provide the best experience for both shoppers and admin. It implements the minimalist yet elegant that allows even new users to quickly know how to navigate around on their first visit. For admins, the system is designed in such a way that it will prevent mistakes from deleting or editing the wrong item with a confirmation check. Overall, the system also has many errors handling techniques to prevent the system from crashing and it can run smoothly. For UI design, please refer to the screenshot section below.

# Deployment

The frontend of the system is live and be accessed via: <https://hoangdesu.github.io/Kiti-fullstack-React-SpringBoot/>

It would require the backend running locally with a database to fully fetch data.

# System design

localhost:5678

localhost:3456

Kiti’s frontend  
(React)

Kiti’s backend  
(Spring Boot)



User   
(shoppers/admin)

localhost:8765

Database

(Postgres)

# Backend: Admin’s dashboard

Graphical user interface

Description automatically generated

Figure 1 - Admin login page

Admin’s inventory management dashboard

Table

Description automatically generated

Add new product form

Graphical user interface, application

Description automatically generated

Delete confirmation modal

Graphical user interface, application, table

Description automatically generated

# Frontend: Kiti’s storefront

Graphical user interface, website

Description automatically generated

User can search for an item from a product name or other information related to that product

Graphical user interface, application

Description automatically generated

User can filter by categories

Graphical user interface, text, application

Description automatically generated