**Specification bill of Super Jahiz**

**Introduction**

SuperJahiz is an ecommerce platform that consists of selling ecommerce websites as a SaaS model, while the full product is to be deployed into the market, A part of the project will be presented as a school project for the JEE module.

In the full version, there will be two parts, the website / web app that is the portal to the service, where clients can subscribe to gain access to their own ecommerce website, the second part is the ecommerce websites themselves, which will be what the school project revolves around, the school project will contain the MVP features of the ecommerce template.

For this specification bill version, only school project related features will be mentioned.

**Features**

**Admin panel:**

* Authentication with email and password.
* CRUD operations on products and labels(categories) (a product can have one label, a label can be associated with many products).
* A visualization dashboard containing the total of sales, deliveries that need to be sent, products that are out of stock, and the total net value of the stock currently existing, as well as a graph showing the sales during the last 7 days, (other statistics may be added if there is enough time).
* Options related to the visual aspect of the website (like the accent color).

**E-commerce platform**

* Registration requires a username, an email, a full name, a phone number, and a password.
* A display of the list of products where each product has a name, a description, a price, a label, the quantity currently in stock, and the units sold.
* (if there is enough time), a real time search and filter option (filter by price range…)
* A cart where all selected products are added, quantities can be changed:
  + A product can be removed and the total is calculated (the payment system will not be included in the school project as it will not be handled in JEE (will be out-sourced).
  + When the order is passed, the cart is emptied and transformed into an order which can be visualized by the user and the admin.
  + An order has a list of products, the quantity of the products, the total price, the status (pending, shipping, shipped) and the user attached to it.
  + The admin can change the status of the order (manually for now), the status can only be moved from pending to shipping, then from shipping to shipped.
  + When the status changes to shipping, the amount of products contained is deducted from the stock, the status can only be changed to shipping if the amount of products in an order are all inferior or equal to the existing stock.
  + When the status passes to shipped, the total amount of the order is added to the profit of the day of the status change, then for each product, the amount sold will be incremented accordingly (that can be just calculed through foreign keys but adding it as a field will create little redundancy, and will simplify the process).

**Technologies used**

This project will follow a server-client architecture:

* The server side will be a RESTful API using JEE.
* The client will be a web app using ReactJs.
* (as of the moment of writing this document, the library to be used for the linking has not been decided, but either Axios, fetch, or react query will be used unless a better alternative is found).
* MySql will be the database of choice, (MongoDb might be implemented as a hybrid in the case where raw JS objects need to be stored).
* Authentication for both the admin and regular users will be handled via JWTs.