



# MegaStore Marketplace

**By Adam Khomsi, Youssef Hajjari,**

**Youssef El Bakkouri et El Mahdi Rammach**

YouCode Safi

Second year, Robert Noyce

Javascript

2021 - 2022

# 1. Project description:

MegaStore is an online platform marketplace where users can become sellers and create their store and manage it to sell them on the platform.

Once a user creates a store, it waits to be validated by the super admin.

Super admin can validate sellers and create admins.

The admins manage the client orders.

Sellers can add products to their store once validated.

The user as a client can browse the marketplace and place the products he wants to buy in his cart before performing a checkout.

# 2. Technologies used:

## 2.1. Software design:

**draw.io:** A general-purpose modeling language free software that visualizes the design of a system through diagrams.

## 2.2. UI and UX design:

**Ali Express:** based on the new Ali Express design.

## 2.3. Software development:

**Visual Studio Code:** Integrated Development Environment (IDE) using many extensions such as:

- **draw.io:** draw.io integration in VS Code.
- **Thunder Client:** lightweight Rest API Client Extension.
- **GitLens:** Lets you glimpse into whom, why, and when a line or code block was changed, then jump back through history to gain insights on the code's evolution.

**Database:**

- **MongoDB:** is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.
- **Mongoose**: is a JavaScript object-oriented programming library that creates a connection between MongoDB and the Express web application framework.

**Backend:**

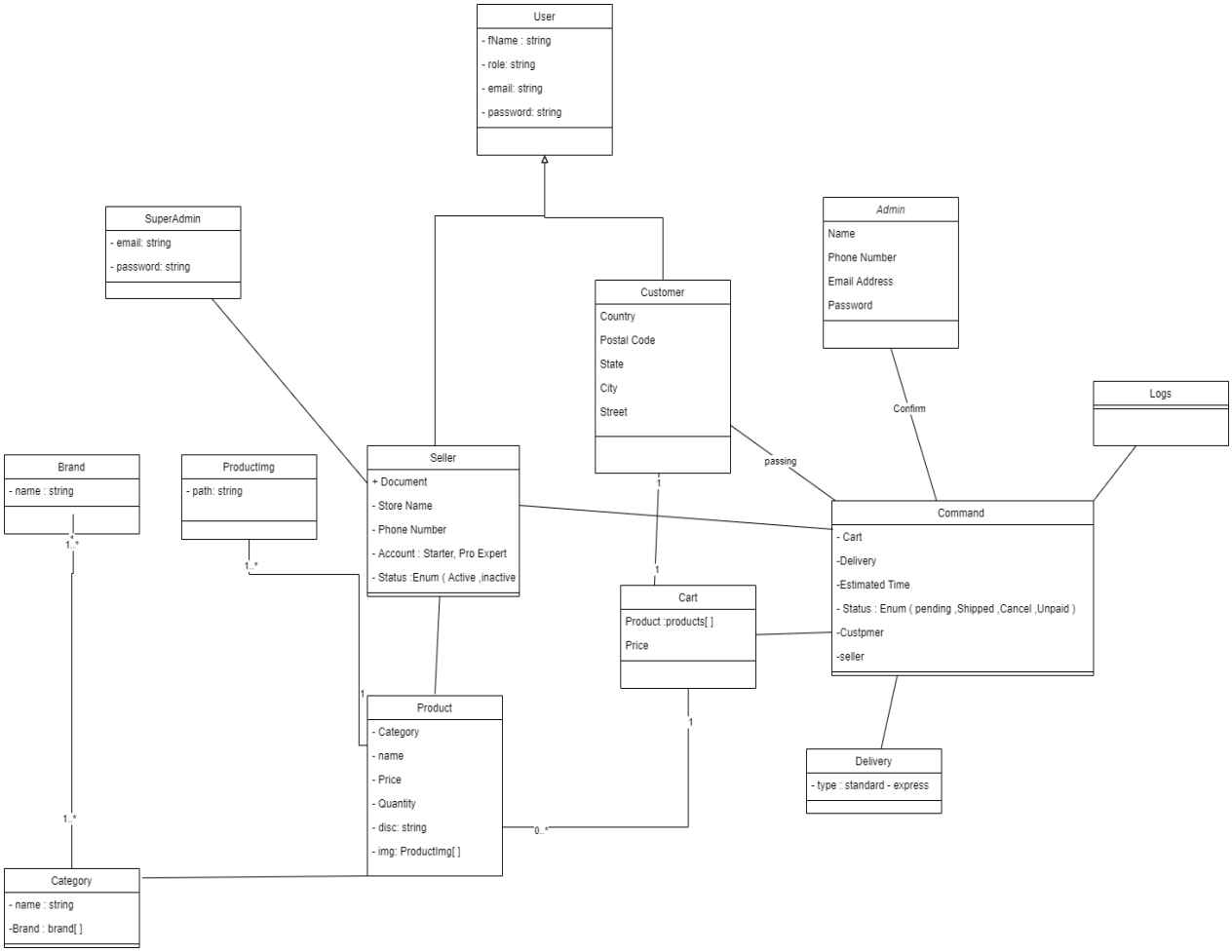
- **Node.js:** JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser.
- **Express.js:** Web application open-source framework for Node.js. It is designed for building web applications and APIs.
- **GraphQL:** Open-source data query and manipulation language for APIs.
- **Apollo Server:** Open-source, spec-compliant GraphQL server that's compatible with any GraphQL client, including Apollo Client.

**Frontend:**

- **Next.js**: Open-source web development framework built on top of Node.js enabling React based web applications functionalities such as server-side rendering and generating static websites.
- **TailwindCSS:** Utility-first CSS framework for rapidly building custom user interfaces. It is a highly customizable, low-level CSS framework that gives you all of the building blocks you need to build bespoke designs without any annoying opinionated styles you have to fight to override.

- **Apollo Client:** Comprehensive state management library for JavaScript that enables you to manage both local and remote data with GraphQL.

# 3. UML Class diagram:



## 4. Development process:

After finishing the software design class diagram with UML, we began working on the database setup (MongoDB on mongo Atlas). After that we started configuring the backend server in Express and layed out the architecture and structure

We picked GraphQL as our query language since it allows us to do complex queries for our use case.

We also chose to work with TypeScript over plain JavaScript, since it allows us to check for any errors at compile time instead of run time and on top of types checking and various features.

For the frontend, we used Next js which allows server side rendering and SEO support while working with React.

The tasks were divided between us on a Kanban board in Trello.

## 5. Conclusion

We learned a lot from this project while working and trying new technologies. Trying to coordinate despite a lot of hardships along the way.